

**Escola Naval**

**DEPARTAMENTO DE FORMAÇÃO DE ENGENHEIROS  
NAVAIS RAMO DE ARMAS E ELETRÓNICA**



***Otimização das Standard Operator Checks (SOCS)- Utilização de  
Dispositivos Móveis na Execução das SOCS***

**Adilson Euclides Maquita António**

**MESTRADO EM CIÊNCIAS MILITARES NAVAIS**

**2013**



**ESCOLA NAVAL**  
**DEPARTAMENTO DE FORMAÇÃO DE ENGENHEIROS NAVAIS**  
**RAMO DE ARMAS E ELETRÓNICA**



***Otimização das Standard Operator Checks (SOCS) - Utilização de  
Dispositivos Móveis na Execução das SOCS***

O Mestrando,  
*(assinado no original)*

---

ASPOF Maquita António

O Orientador,  
*(assinado no original)*

---

CFR EN-AEL Ribeiro Correia

O Coorientador,  
*(assinado no original)*

---

CTEN M Dias Marques



## **DEDICATÓRIA**

Dedico este trabalho à minha esposa que sempre me apoiou nos momentos difíceis durante a minha formação, pelos sacrifícios que teve de fazer em prol da nossa família, para que fosse possível a concretização deste trabalho.

Obrigado por tudo.



## AGRADECIMENTOS

A realização da presente dissertação muito deve aos contributos e aos incentivos de várias entidades às quais gostaria de dirigir algumas palavras de gratidão, em particular:

À Escola Naval, pela formação militar e académica que me foi prestada;

Ao meu tutor Engenheiro Ribeiro Correia e ao Cotutor Capitão Tenente Dias Marques, pelo incentivo, pela indicação do melhor caminho a seguir na elaboração da tese, que me fizeram compreender alguns conceitos e ver outros com outra perspetiva. Sem a vossa ajuda, este trabalho não teria chegado onde chegou;

Ao Capitão Tenente Cortes Sanches e à Primeiro Tenente Isabel Bué pelos valiosos contributos no âmbito da presente investigação;

À camarada Ganança do Carmo pelo bom trabalho de equipa que tivemos durante o desenvolvimento das aplicações, e nas entrevistas feitas aos oficiais;

Aos camaradas do 5º ano da Classe de Engenheiros Navais do ramo de Armas e Eletrónica pela camaradagem e pelas discussões e críticas construtivas que tivemos em relação a tese;

Aos camaradas dos Países Africanos de Língua Oficial Portuguesa e aos camaradas da república da Argélia um especial obrigado pela amizade;

À toda minha família e todos meus amigos de verdade.

À minha esposa em especial, pelo apoio e paciência que teve no decorrer deste último ano de formação, obrigado pela confiança depositada em mim, “és a minha heroína”.

Obrigado!





## RESUMO

A prática de execução das SOC's na preparação do navio antes da largada é iminentemente móvel e com necessidades de documentação substanciais, no que concerne aos registos das mesmas. No entanto, os sistemas de informação utilizados atualmente, não estão aptos para fazer a atualização do estado dos equipamentos em tempo real, sendo essa tarefa adiada para momento posterior.

A evolução dos dispositivos móveis, associado o desenvolvimento das capacidades para operação em rede sem fios, promove novas formas de os utilizar no auxílio à execução das SOC's. Assim, esses dispositivos podem ser utilizados para estenderem a informação para todos pontos no navio, possibilitando assim uma melhor partilha de informação em tempo real. Para além disso, a tecnologia móvel e as capacidades emergentes introduzem novas possibilidades, principalmente no que concerne a interatividade, entre o utilizador e aplicação móvel instalada no dispositivo.

Nesta dissertação foi desenvolvido um sistema de informação integrado para apoiar o utilizador no registo dos itens da SOC de cada equipamento. Este sistema surge com a intenção de utilizar a tecnologia móvel para otimizar o modo de execução das SOC's atuais. Além da tecnologia móvel, o sistema também tem um módulo web que permite o acompanhamento da execução através de um computador, permitindo assim que todos os elementos do navio tenham acesso a informação. Também é possível armazenar os registos de cada atividade na base de dados do servidor, o que permite uma posterior avaliação das atividades decorrida em cada SOC, de acordo com a data da realização da mesma e o elemento responsável pela mesma.

Palavras-chave: dispositivos móveis, sistemas de informação, base de dados, rede sem fio.



## **ABSTRACT**

The practical implementation of SOC's in the preparation of the ship before departure is imminently movable and there are necessities of substantial documentation in relation to records of the same one. However, the information systems used currently are not able to update the status of devices in real time, which postpone this task to later time.

The evolution of mobile devices, coupled with the development of capabilities for operation in wireless network, promotes new ways to use them to help in the implementation of SOC's. Thus, these devices can be used to extend the information to all points on the ship, thus enabling better sharing of information in real time. In addition, mobile technology and emerging capabilities introduce new possibilities, especially with regard to interaction between the user and application installed on the mobile device. In this thesis we have developed an integrated information system to support the user in the registration of items of the SOC in each equipment. This system comes with the intention to use mobile technology to optimize the execution mode of the current SOC's. Besides the mobile technology, the system also has a web module that allows the monitoring of implementation through a computer, allowing all elements of the ship have access to information. It is also possible to store the records of each activity on the database server, allowing further evaluation of the activities elapsed in each SOC, according to the date of the same ones and the element responsible for the same activities.

**Keywords:** mobile, information systems, database, wireless network.



## ÍNDICE

DEDICATÓRIA .....	i
AGRADECIMENTOS .....	iii
RESUMO .....	v
ABSTRACT .....	vii
LISTA DE FIGURAS .....	xiii
LISTA DE TABELAS .....	xv
LISTA DE GRÁFICOS.....	xvii
LISTA DE SIGLAS E ACRÓNIMOS .....	xix
GLOSSÁRIO.....	xxiii
CAPÍTULO I INTRODUÇÃO.....	1
1.1. Enquadramento geral .....	1
1.2. Problemática e Importância do Estudo .....	1
1.2.1. SOCs Atuais .....	2
1.2.2. Solução Proposta .....	4
1.3. Organização da Tese .....	6
CAPÍTULO II INTERFACE COM UTILIZADOR .....	7
2.1. Introdução .....	7
2.2. Aplicação Web.....	8
2.2.1. Protocolo.....	9
2.3. Aplicações Móvel .....	12
2.3.1. Plataformas .....	13
2.4. Dispositivos .....	14
2.4.1. <i>Smartphones</i> .....	17
2.4.2. <i>Tablet</i> .....	19
2.4.3. <i>Ruggedized</i> .....	22
2.5. Conclusões .....	23
CAPÍTULO III SERVIÇOS TÉCNICOS .....	25
3.1. Servidores Web.....	25
3.1.1. IIS .....	25

3.1.2.	Apache .....	26
3.1.3.	Apache Tomcat.....	26
3.2.	Tecnologias .....	27
3.2.1.	Introdução .....	27
3.2.2.	JSP .....	28
3.2.3.	ASP.NET .....	29
3.2.4.	PHP .....	31
3.3.	Linguagens.....	31
3.3.1.	XML .....	32
3.3.2.	HTML.....	33
3.3.3.	HTML4.....	33
3.3.4.	AJAX.....	34
3.3.5.	HTML5 .....	35
3.4.	Serviços de Informação e Integração .....	36
3.4.1.	<i>Windows Phone</i> .....	39
3.4.2.	iOS .....	40
3.4.3.	Android.....	40
3.5.	Equipamento de serviços de informação .....	42
3.5.1.	Modelos de Base de Dados.....	42
3.5.2.	Características e Requisitos de um SGBD .....	43
3.5.3.	Access.....	44
3.5.4.	SQL <i>Server</i> .....	45
3.5.5.	MySQL.....	45
3.6.	Serviços de Comunicações .....	46
3.6.1.	Rede Ethernet .....	47
3.6.2.	Redes Sem Fios .....	49
3.7.	Conclusões .....	54
CAPÍTULO IV DESCRIÇÃO DAS APLICAÇÕES .....		59
4.1.	Introdução .....	59
4.2.	Requisitos da Aplicação móvel.....	60
4.3.	Requisitos da Aplicação web .....	65
4.4.	Estrutura do Código Fonte das Aplicações .....	68
4.4.1.	Código fonte da aplicação móvel .....	69

4.4.2. Código fonte da aplicação Web.....	72
4.4.3. Estrutura da Base de Dados .....	74
CAPITULO V TESTES E VALIDAÇÃO .....	79
5.1. Objetivos e condições dos Testes .....	79
5.2. Descrição dos testes realizados .....	82
5.3. Resultados obtidos .....	82
5.3.1. Desempenho .....	84
5.3.2. Viabilidade .....	87
5.4. Limitações e Problemas encontrados.....	88
CAPITULO VI CONCLUSÕES .....	89
6.1. Conclusão.....	89
6.2. Recomendações para trabalhos futuros.....	90
REFERENCIAS BIBLIOGRÁFICAS .....	91
ANEXOS.....	95
Anexo A – Dispositivos móveis com as características adequadas para o projeto.....	95
Anexo B - SOCS utilizadas para o desenvolvimento das Aplicações .....	97
Anexo C – Código fonte das <i>Activities</i> da Aplicação móvel.....	130
Anexo D – Código fonte dos <i>Web Services</i> .....	225
Anexo E – Código fonte da Aplicação Web.....	238
Anexo F – Fichas de Testes das Aplicações .....	282





## LISTA DE FIGURAS

Figura 1 - Arquitetura do protótipo .....	7
Figura 2 - Processo de comunicação HTTP .....	9
Figura 3 - Exemplo de mensagem de pedido HTTP .....	10
Figura 4 - Exemplo de mensagem de resposta HTTP .....	10
Figura 5 - Telefone Motorola DynaTAC 8000X.....	15
Figura 6 - SmartPhone Samsung Galaxy S .....	17
Figura 7- iPad da Apple.....	20
Figura 8 - Tablet Híbrido Lenovo IdeaPad.....	21
Figura 9 - Tablet Panasonic ToughPad.....	23
Figura 10 - Arquitetura da plataforma Android.....	42
Figura 11 - Representação da interface entre a rede e as aplicações .....	47
Figura 12 - Padronização global IEEE para redes sem fio .....	51
Figura 13 - Modos de configuração básicos na tecnologia Wi-Fi.....	53
Figura 14 - Diagrama do sistema a implementar.....	59
Figura 15 - Diagrama dos Equipamentos a Implementar na Aplicação .....	61
Figura 16 - Menu Principal da aplicação móvel.....	62
Figura 17 - Layout da Lista de Equipamentos do Oficial de quarto à ponte.....	63
Figura 18 - Exemplo do Corpo da SOC .....	64
Figura 19 - Página principal das SOC's.....	66
Figura 20 - página de acompanhamento da SOC do Radar Kelvin Hughes.....	67
Figura 21 - Página para a consulta do histórico.....	67
Figura 22 - Estrutura de uma aplicação android.....	69
Figura 23 - <i>Activities</i> criadas para o desenvolvimento da aplicação móvel .....	71
Figura 24 - Lista dos <i>Web Services</i> utilizados no desenvolvimento da aplicação.....	72
Figura 25 - Páginas web criadas para aplicação Web .....	74
Figura 26 - Tabelas da Base de Dados .....	77
Figura 27 - Tabela com o Registo dos tempos realizados em cada SOC .....	78



## LISTA DE TABELAS

Tabela 1 - Comparação entre as SOC's atuais e a Solução a Implementar .....	5
Tabela 2 - <i>Smartphones</i> vendidos no mundo, em milhões de unidades. ....	19
Tabela 3 - <i>Tablets</i> vendidos no mundo, em milhões de unidades .....	21
Tabela 4 - Volume de <i>Smartphones</i> vendidos com Sistema Operativo específico, a nível mundial, em milhões de unidades.....	38
Tabela 5 - Características de algumas variantes de Ethernet.....	49
Tabela 6 - Variantes da norma IEEE 802.11 .....	52
Tabela 7 - Comparação de Sistemas Operativos Móveis .....	56
Tabela 8 - Procedimentos de Teste da Aplicação Móvel .....	81
Tabela 9 - Procedimento de Teste da Aplicação Web.....	81
Tabela 10 - Tabela do tempo padrão definido no SIMNAV .....	83
Tabela 11 – Comparação do registo de tempo efetuado no SIMNAV com o tempo padrão .....	84
Tabela 12 - Comparação do registo de tempo efetuado no Navio com o tempo padrão.....	84
Tabela 13 - Comparação do registo de tempo efetuado no Navio com o tempo padrão (continuação) .....	85
Tabela 14 - Comparação de desempenho entre os sistemas.....	87
Tabela 15 - Erros e Limitações encontrados durante os testes .....	88



## LISTA DE GRÁFICOS

Gráfico 1 – Mercado mundial de vendas de <i>smartphone</i> , entre o terceiro trimestre de 2011 e o terceiro trimestre de 2012. ....	18
Gráfico 2 - Quota do mercado mundial de sistemas operativos móveis para smartphones, entre o terceiro trimestre de 2011 e o terceiro trimestre de 2012. ....	38



## LISTA DE SIGLAS E ACRÓNIMOS

AJAX	Assynchronous JavaScript and XML
ANSI	American National Standards Institute
AP	Access Point
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ASPOF	Aspirante á Oficial
BREW	Binary Runtime Environment for Wireless
BYOD	Bring Your Own Device
CERN	Conseil Européen pour la Recherche Nucléaire
CIS	Communication and Information System
CITAN	Centro Integrado de Treino e Avaliação Naval
CSMA	Carrier Sense Multiple Access
CSMA/CD	Carrier Sense Multiple Access with Collision Detect
CSS	Cascading Style Sheets
DOM	Document Object Model
DTD	Document Type Definition
EN	Escola Naval
EN-AEL	Engenheiro Naval no ramo de Armas e Eletrónica

FOP	Fichas de Verificação do Oficial de Quarto-à-Ponte
GPS	Global Position System
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IDC	International Data Corporation
IEEE	Institute of Electrical and Electronic Engineer
IIS	Internet Information Services
JSP	Java Server Pages
LCD	Liquid Crystal Display
MAC	Media Access Control
MIME	Multipurpose Internet Mail Extensions
MIMO	Multiple Input, Multiple Output
MMS	Multimedia Messaging Service
MTU	Maximum Transmission Unit
NATO	<i>North Atlantic Treaty Organization</i>
NRP	Navio da República Portuguesa
ODBC	Open Data Base Connectivity
PDA	Personal Digital Assistants
PHP	Hypertext Preprocessor



SDK	Software Development Kits
SGBD	Sistema de Gestão de Base de Dados
SIMNAV	Simulador Naval
SMS	Short Message Service
SO	Sistema Operativo
SOAP	Simple Object Access Protocol
SOC	Standard Operator Checks
SQL	Structed Query Language
SSID	Service Set Identifier
SSL	Secure Sockets Layers
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
USB	Universal Serial Bus
VB	Visual Basic
WECA	Wireless Ethernet Compatibility Alliance
WHATWG	Web Hypertext Application Technology Working Group
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Area Network

XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language
XSLT	eXtensible Style Language Transformation

## GLOSSÁRIO

<i>Activity</i>	É uma janela que contém a interface do utilizador duma determinada aplicação em Android
<i>API</i>	Conjunto de rotinas estabelecidas por um <i>software</i> para a utilização das suas funcionalidades por aplicações
<i>ASCII</i>	É uma codificação de caracteres de sete bits baseada no alfabeto inglês
<i>Bluetooth</i>	Tecnologia de comunicação sem fios que interliga e permite a transmissão de dados entre dispositivos eletrónicos através de ondas de rádio
<i>Browser</i>	Programa que habilita seus utilizadores a interagirem com documentos virtuais da Internet, também conhecidos como páginas da web
<i>Code behind</i>	Técnica de separação do código em dois ficheiros: um, em HTML e ASP.NET para interface gráfica; outro, em VB.NET ou C# para o código que confere as funcionalidades a essa interface.
<i>Fading</i>	Desvio da atenuação que um sinal de telecomunicação de frequência modulada pelo portador experimenta sob certos meios de propagação.
<i>Framework</i>	Um conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação. Em desenvolvimento de <i>software</i> , é uma abstração que une códigos comuns entre vários projetos.
<i>Hardware</i>	É toda a parte física que constitui o dispositivo eletrónico.

<i>Hotspot</i>	Nome dado aos locais públicos onde há cobertura do serviço de comunicações, normalmente internet, sem fio através da tecnologia Wi-Fi
<i>Java EE</i>	É uma plataforma de programação para servidores na linguagem de programação Java.
<i>Malware</i>	<i>Malicious software</i> é um software destinado a se infiltrar num sistema alheio de forma ilícita
<i>Middleware</i>	Designa camadas de <i>software</i> que não constituem aplicações, mas facilitam o uso de ambientes ricos em tecnologias de informação
MMS	Significa serviço de mensagens multimédia. Tecnologia que permite a transmissão de texto, imagem, áudio e vídeo entre telemóveis ou entre telemóvel e correio eletrónico
ODBC	Norma que define um conjunto de interfaces que permitem o uso de linguagens de programação capazes de utilizar estas interfaces, para ter acesso a uma vasta gama de bases de dados distintas sem a necessidade de codificar métodos de acesso especializados.
<i>Open Source</i>	Significa que é permitido desenvolver outros tipos de conceito semelhante com base nele
<i>Ruggedized</i>	Termo inglês utilizado para mencionar uma versão mais robusta de alguma coisa, de modo a que esta tenha capacidade de suportar ambientes hostis, tem maior durabilidade que a versão normal.

<i>Scriptlet</i>	Utilizado na tecnologia JSP, é um pedaço de código Java embutido no código JSP.
Servidor web	<i>Software</i> que é executado no computador servidor responsável pela hospedagem dos websites. Com objetivo de disponibilizar as paginas web
<i>Servlet</i>	Módulo de extensão da funcionalidade de um servidor Web, através de aplicações implementadas em Java, também pode ser considerado como uma classe na linguagem de programação Java
<i>Smalltalk</i>	É uma linguagem de programação orientada a objeto dinamicamente.
SOAP	É um protocolo para troca de informações estruturadas em XML
SOC	Verificação da operacionalidade dos equipamentos dos navios, antes da largada ou em procedimentos de rotinas.
<i>Software</i>	É a parte lógica cuja função é fornecer instruções para o <i>hardware</i> . Estas instruções são interpretadas por um aparelho eletrônico com o objetivo de executar tarefas específicas
<i>Standby</i>	Significa “em espera”. Termo utilizado em aparelhos eletrônicos para definir que o aparelho está ligado mas não está a ser utilizado, economizando assim energia.
Tuplos	Representação de cada linha de uma base de dados, formada por um conjunto de colunas

USB	É o protocolo de comunicação que suporta transferência de dados entre computadores e periféricos.
<i>Wireless</i>	Significa rede sem fios. Refere-se a uma rede de dispositivos sem a necessidade de uso de cabos, as comunicações são feitas via ondas de rádio ou via infravermelhos.
Wireshark	Programa que analisa o tráfego de rede, e organiza por protocolos, permite saber os dados que entram e/ou saem do computador ou da rede que computador esteja ligado.

# CAPÍTULO I INTRODUÇÃO

## 1.1. Enquadramento geral

A introdução das novas tecnologias nos mais diversos locais de trabalho é hoje uma realidade. Existindo áreas de maior divulgação, tais como gestão e contabilidades, e outras em que a sua penetração não é tão grande tal como acontece no aprontamento dos navios da Marinha Portuguesa, no que concerne a verificação do estado de operacionalidade dos equipamentos.

Os elementos do navio responsáveis pela execução das SOC's têm um trabalho muitas vezes móvel. Deslocam-se aos pontos de aviso áudio para o navio e necessitam de criar um conjunto elevado de documentação, com a informação relativa a cada equipamento. Os sistemas de informação, tipicamente, não acompanham o profissional, o que leva a adiamentos de registo e inevitavelmente a falhas na documentação por esquecimento.

Deste modo, dado o avanço das tecnologias *wireless* e dos dispositivos móveis, é possível introduzir estas tecnologias nos navios, e a sua integração com os sistemas de informação já existentes nos mesmos. Esses dispositivos permitem a disponibilização de informação de forma facilitada no navio. Permitem também, consultas de histórico minimizando assim o risco de serem cometidos os mesmos erros, pois é possível analisar os dados fornecido pelo histórico.

Baseado nessa revolução tecnológica surgiu a ideia de criar um sistema que auxilie os utilizadores na execução das SOC's em ambiente móvel. Assim, numa primeira abordagem a esse conceito, foi criado um sistema que permite os elementos de quarto à ponte executarem as SOC's com auxílio de um dispositivo móvel e a permitir a atualização dos dados em tempo real, permitindo assim a consulta dos mesmos por outros elementos do navio.

## 1.2. Problemática e Importância do Estudo

É importante que uma organização como a Marinha tenha um sistema, quer em papel ou não, que permita verificar a operacionalidade dos equipamentos utilizados pelos seus navios.

Atualmente, as SOC's que estão disponíveis na ponte dos navios da Marinha, apresentam-se num formato em papel e são preenchidas pelos oficiais e outros elementos de quarto à ponte. Existem apenas alguns registos manuais, sobre o estado dos equipamentos de navegação na ponte no que respeita à execução das ações previstas nas SOC's, a informação sobre estas ocorrências pode, por vezes, demorar a chegar aos destinatários de interesse. Esta realidade dificulta a avaliação posterior das ações efetuadas e afeta a eficiência e a eficácia de todos os envolvidos. Além disso, torna muito difícil a identificação dos pontos críticos de modo a que estes possam ser corrigidos posteriormente.

Propõe-se assim, implementar uma aplicação informática, baseada em dispositivos móveis e redes sem fios que permita:

- a) **Uma introdução dos eventos e informação associada de interesse;**
- b) **Viabilize a existência de um histórico de registos com maior precisão;**
- c) **Disponibilize no momento certo a informação necessária a quem dela necessite.**

Com o objetivo de otimizar as atividades da ponte com o auxílio das novas tecnologias.

### **1.2.1. SOC's Atuais**

O modo como são realizadas as SOC's atualmente, não evidenciam o rendimento dos elementos que as utilizam. Não existe supervisão, ou um registo através do qual os elementos da guarnição responsáveis pela execução dos testes aos equipamentos possam saber qual a SOC que está a ser realizada de acordo com a ordem correta de prioridades.

Além disto, não existe um histórico sobre a realização das SOC's que permita analisar o seu tempo de execução ou outros detalhes de execução, impossibilitando assim que se analise, por exemplo, o porquê das SOC's de um mesmo equipamento demorarem mais tempo a ser realizadas por alguns elementos da guarnição do que por outros.

O controlo do tempo também é um dos problemas, visto que não há um controlo efetivo que permita tirar ilações, sobre o tempo de execução (se estamos dentro ou fora do tempo padrão), pois de acordo com Vieira (2012), as SOC's devem ser realizadas 45min antes da largada do navio, sendo este o único tempo de referência a cumprir. Isto leva a que seja difícil perceber em que pontos específicos os elementos responsáveis falham, ou ocupam algum tempo, ou se o problema é mesmo do equipamento.



Apesar do sistema utilizado apresentar debilidades, é possível encontrar pontos fortes do mesmo, tais como:

- Listas plastificadas: o facto de as listas serem plastificadas, torna-as bastante resistentes a água e sujidade;
- Facilidade de reutilização: dado que uma lista pode ser escrita, dobrada, apagada (os dados escritos) e reescrita sem colocar em causa a sua capacidade operacional.
- Portabilidade: São fáceis de transportar e o utilizador pode tê-la em sua posse como um “cartão de detalhe”;
- Maleabilidade;
- Independência Energética: As SOC's atuais são independentes de fontes energéticas.

Os pontos fracos encontrados nas SOC's devem-se, sobretudo, por ser um documento em papel, e apresentam os seguintes inconvenientes:

- Maior ocupação do espaço físico: visto que para cada equipamento tem de haver uma SOC em papel. Dada a quantidade de equipamentos existentes num navio, o número de SOC's tem de ser equivalente;
- Manter os Arquivos: torna-se problemático quando existe um elevado número de documentos, pois a dimensão de espaço ocupado cresce;
- Documentos Estáticos: Os documentos estáticos não alertam o utilizador quando os tempos de operação são ultrapassados; não contabilizam o tempo de cada atividade nem o tempo geral; não mostram uma imagem relativa ao item a ser verificado; Não enviam a informação logo de seguida a execução de um item, de uma determinada SOC;
- Transmissão de informação automática: As listas de verificação em papel não permitem uma transmissão de informação automática para outros pontos do navio, sendo essencial a comunicação por voz para indicar que a SOC foi concluída com sucesso, ou não;

Devido à capacidade nula de interação com o utilizador, as listas não servem como um sistema de apoio à decisão, apenas têm a função de auxiliar a equipa para que esta se certifique que todas as tarefas essenciais foram realizadas.

### 1.2.2. Solução Proposta

Propõe-se assim implementar uma aplicação informática, baseada em dispositivos móveis e redes sem fios. Esta aplicação vai permitir uma introdução dos eventos e informação associada de interesse, que viabilize a existência de um histórico de registos, com maior precisão, e que disponibilize no momento certo, a informação a quem dela necessite. Além disto, não há constrangimento relativamente ao espaço ocupado, pois um documento digital, ao contrário do documento em papel, cinge-se ao *hardware*<sup>1</sup> que o contém. Para determinar a adequabilidade dos sistemas informáticos que serão implementados, considerou-se que a solução proposta será capaz de resolver alguns dos problemas encontrados nas SOC's em papel. Além disso, identificou-se os pontos fracos que este sistema possui de modo a garantir que estes não se sobrepõem aos seus contributos.

Ora, Um dos aspetos positivos da aplicação a desenvolver é a cronometragem automática dos tempos de execução de cada atividade, e no final da execução da SOC apresentará o tempo total de execução. Outro aspeto a salientar é que no final a aplicação apresentará o estado atual do equipamento, caso este esteja operacional ou não.

A característica principal documentos digitais é a sua desmaterialização, ou seja, todas as SOC's estão no mesmo sistema. O volume de armazenamento reduzido, a pesquisa inerente, alteração do documento original são também características únicas dos documentos digitais. O Sistema de Informação e Comunicação (CIS)<sup>2</sup> proposto irá conter a informação tratada, guardada na base de dados e disponibilizada na aplicação Web, e também a aplicação que contem as SOC's dinâmicas disponibilizadas no dispositivo móvel.

Desta forma, será possível corrigir as falhas verificadas nas SOC's em papel, nomeadamente, no caso da aplicação para dispositivos móveis, onde será garantido o registo e acesso aos eventos de interesse e informação associada; a informação sobre tempos de execução das várias atividades. A aplicação web, para os restantes elementos do navio, permitirá o acesso ao estado das atividades em curso; fornecerá informação sobre tempos de execução das várias atividades; permitirá analisar o desempenho das várias execuções de uma SOC (o que corre bem, o que corre mal, etc.).

Apesar desta nova apresentação das SOC's ter muitos aspetos positivos, a mesma tem aspetos negativos. Contrariamente aos documentos em papel, os documentos digitais dependem dum ambiente tecnológico específico que permita o acesso e a interpretação dos

---

<sup>1</sup> *Hardware* é toda a parte física que constitui o dispositivo eletrónico

<sup>2</sup> CIS, acrónimo para Communication and Information System .

seus conteúdos. A informação digital é criada por um *software*<sup>3</sup>, que corre num dispositivo com determinadas características tecnológicas. A dependência do ambiente tecnológico e fontes de energia são as maiores ameaças e as maiores vulnerabilidades dos documentos digitais. Além disso, os *softwares* estão em constante atualizações que os tornam mais capazes e, ao mesmo tempo, rapidamente obsoletos a nível técnico e comercial. O *hardware* é também mais suscetível à deterioração e degradação física, quando comparados a uma folha de papel plastificada.

	<b>SOCs atuais</b>	<b>SOCs dinâmicas</b>
<b>Pontos Fracos</b>	Maior ocupação do espaço físico; Manter os Arquivos; Documentos Estáticos; Transmissão de informação automática;	Dependência dum ambiente tecnológico específico; Dependência de fontes de energia; Deterioração e degradação física;
<b>Pontos Fortes</b>	Independência Energética Facilidade de reutilização Maleabilidade; Listas plastificadas.	Cronometragem automática; Desmaterialização; Interatividade; Utilização de computador (na aplicação web) Transmissão de informação automática.

**Tabela 1 - Comparação entre as SOC's atuais e a Solução a Implementar<sup>4</sup>**

<sup>3</sup> *Software* é a parte lógica cuja função é fornecer instruções para o *hardware*. Estas instruções são interpretadas por um aparelho eletrónico com o objetivo de executar tarefas específicas

<sup>4</sup> Foi feita uma análise Swot para poder detetar os pontos fortes e fracos de cada um dos sistemas.

### 1.3. Organização da Tese

A presente dissertação está organizada em seis capítulos distintos, no intuito de evidenciar todo trabalho efetuado durante o desenvolvimento da mesma.

No capítulo I é efetuado o enquadramento geral ao tema de estudo, além disto, também é definido o método de investigação utilizado para este estudo, no qual é referido as etapas que foram seguidas para obtenção do produto final.

No capítulo II e capítulo III é efetuada a definição do estado da arte tecnológico a implementar, tendo como base a arquitetura protótipo da dissertação. O capítulo II aborda aspetos relacionados com a revisão do estado da arte sobre aplicações moveis e aplicações web, dando enfase aos protocolos utilizados nas web e também as plataformas de desenvolvimento para aplicações moveis referenciando os dispositivos alvos. Já o capítulo III trata de aspetos relacionados com os tipos de servidores web a implementar, bem como as tecnologias de desenvolvimento para *panas web* e os tipos de linguagens que estão relacionadas com as mesmas. Além disto, também são referenciados os tipos de sistemas gestores de base de dados, e finalmente também é descrito os serviços de comunicações, nomeadamente as rede *ethernet* e a rede sem fio. Convém salientar que ambos os capítulos também constam da dissertação da **ASPOF EN-AEL Ganança do Carmo**.

No capítulo IV é definido os requisitos (na ótica do utilizador) das aplicações desenvolvidas neste projeto, ambos analisados separadamente. É descrito os elementos responsáveis por executar cada uma das atividades e como as aplicações poderão auxiliá-los na execução das mesmas. É também feita uma breve descrição do código das aplicações, especificando assim as classes utilizadas na programação e a estruturação da base de dados utilizada no projeto.

No capítulo V são abordados os testes efetuados a fim de perceber se a solução proposta é realmente uma mais-valia ou não, de acordo com o *feedback* dos utilizadores, também convém salientar que os objetivos traçados para a realização dos testes foram definidos conjuntamente com a camarada **ASPOF EN-AEL Ganança do Carmo**.

No último capítulo (capítulo VI) são feitas as conclusões de todo o trabalho desenvolvido, indicando também as limitações encontradas e algumas sugestões para trabalho futuro.

## CAPÍTULO II INTERFACE COM UTILIZADOR

### 2.1. Introdução

A evolução tecnológica a que se tem assistido na atualidade, tem sido sempre no sentido de se arranjam formas de diminuir o tempo gasto na realização das tarefas, ou então de automatizar essas tarefas, tirando o maior proveito das mesmas. Para atingir o objetivo final deste trabalho – otimizar as atividades da ponte com o auxílio das novas tecnologias – apresenta-se na figura 1, a arquitetura do protótipo que serviu de orientação para decidir a arquitetura alvo da presente dissertação.



Figura 1 - Arquitetura do protótipo<sup>5</sup>

Segundo a doutrina adotada pela NATO<sup>6</sup>, um Sistema de Comunicação e Informação é um agregado de termos de sistemas de comunicação e sistemas informação. Sistemas de comunicação são sistemas de transferência de dados entre pessoas e equipamentos. Sistemas de informação são conjuntos integrados de componentes para a recolha, armazenamento e processamento de dados para transmitirem informação, conhecimento e produtos digitais.

<sup>5</sup> Baseado em (Allied Command Transformation, 2012).

<sup>6</sup> Acrónimo de *North Atlantic Treaty Organization*.

O CIS da arquitetura supramencionada é composto por uma camada de Interface do Utilizador, que contém os requisitos necessários para a interação entre o utilizador final e a informação disponibilizada pelo CIS. Nesta camada estão as Aplicações Web e Móvel, bem como os dispositivos móveis que podem ser utilizados.

Ambas as aplicações acedem à camada de Serviços Técnicos, que tem os Serviços Centrais e os Serviços de Comunicações. Os Serviços Centrais são os responsáveis por disponibilizar os Serviços Web e os Serviços de Informação e Integração. Os Serviços de Comunicações são os sistemas de interligação para transferência de dados entre os vários pontos de acesso. Estes Serviços incluem a Rede Ethernet e Rede Sem Fios. Finalmente, o Equipamento de Sistemas de Informação que se refere ao dispositivo responsável pela hospedagem de *software* que presta auxílio aos Serviços Técnicos, no que diz respeito ao tratamento dos dados operacionais, como é o caso da base de dados.

A interface com o utilizador permite a interação com dispositivos digitais através de elementos gráficos, como ícones e outros indicadores visuais (Ribeiro, 2012). A interação pode ser feita através de um rato, um teclado, ou por toque no caso dos ecrãs táteis. Com qualquer um destes meios, o utilizador é capaz de seleccionar símbolos e manipulá-los de forma a obter algum resultado prático. O ambiente gráfico é um *software* feito para facilitar e tornar prática a utilização de dispositivos através de representações visuais do sistema operativo (Ribeiro, 2012). Este ambiente pode ser criado com recurso a várias linguagens de programação, sistemas operativos, entre outros. Neste capítulo, serão abordados os princípios de funcionamento das aplicações web e móvel, bem como os dispositivos móveis onde as aplicações podem ser disponibilizadas ao utilizador.

Nas secções seguintes serão abordados vários temas relacionados com o desenvolvimento e modelação de aplicações, para além de alguns temas mais genéricos. A análise será feita de duas formas distintas: o desenvolvimento de aplicações móveis e aplicações web e posteriormente os dispositivos móveis.

## **2.2. Aplicação Web**

As aplicações Web são aplicações complexas na medida em que se baseiam numa grande variedade de componentes de *software*, nomeadamente em vários protocolos e linguagens (Miranda, 2010). Ao nível dos serviços, os elementos são organizados da seguinte forma: de um lado está o cliente Web, ou *browser*, que solicita dados ao servidor Web, recebe as respostas, formata a informação e apresenta ao utilizador. Do outro lado,

está o servidor Web que recebe os pedidos, lê os dados escritos em HTML e responde ao cliente (Miranda, 2010).

### 2.2.1. Protocolo

Os protocolos podem ser encarados como uma sequência de algoritmos e regras que regem as trocas de dados nas redes de computadores. O processo de comunicação entre sistemas está organizado em camadas de protocolos. Cada camada protocolar utiliza os serviços fornecidos pela camada ao nível imediatamente abaixo. As camadas protocolares de mais alto nível implementam serviços mais orientados para a aplicação, tais como a transferência de informação hipermídia (Pinheiro, 2012).

### HTTP e HTTPS

HTTP significa *HyperText Transfer Protocol*, e diz respeito a um protocolo que foi criado em 1990 com o objetivo de realizar a transferência de informações entre computadores (W3C, 1996). As transferências de dados na Web baseiam-se numa arquitetura cliente-servidor, em que do lado do servidor está em execução um programa que espera pedidos oriundos do cliente. Esse programa é conhecido como servidor de requisições HTTP (W3C, 1996). A figura 2 mostra o processo de comunicação entre cliente e servidor utilizando o protocolo HTTP.

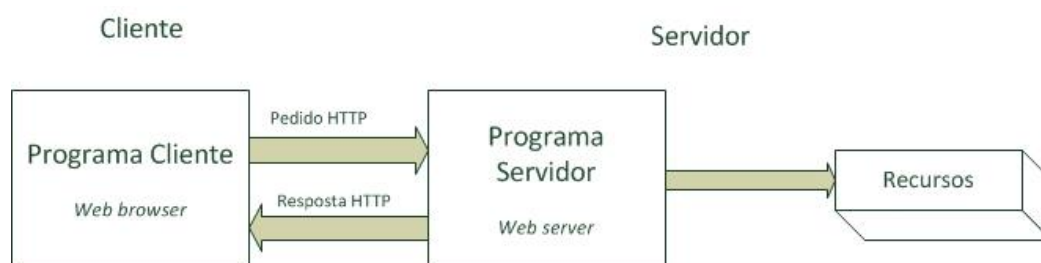


Figura 2 - Processo de comunicação HTTP<sup>7</sup>

### Formatos de mensagens

O HTTP define dois tipos de mensagens: pedido (*request*) e resposta (*response*). Quando há trocas de mensagens entre o cliente e o servidor, estas são escritas em código ASCII<sup>8</sup>, pelo que são legíveis e podem ser produzidas numa interface de linha de comandos para efeitos de teste (Boavida & Bernardes, 2012).

<sup>7</sup> Fonte: Adaptado de Moreira (2005, p.34).

<sup>8</sup> ASCII, é uma codificação de caracteres de sete bits baseada no alfabeto inglês. Cada sequência de códigos na tabela ASCII corresponde a um caractere.

A *request line* inclui a ação a executar sobre a página (método GET), o URL do sítio que está associada a página principal alojada no servidor) e a versão do protocolo que o cliente pretende utilizar. Seguem-se as linhas de cabeçalhos (*headers*) opcionais e o fim de mensagem é assinalado com uma linha em branco identificada pelos caracteres '\r\n' (Kurose & Ross, 2007).

```
HTTP Request Message

GET /somedir/page.html HTTP/1.1
Connection: close
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Accept-language:fr
```

**Figura 3 - Exemplo de mensagem de pedido HTTP<sup>9</sup>**

A mensagem de resposta tem o formato idêntico (ver figura 4), onde em primeiro lugar tem uma linha de estado (*status line*) que inclui o protocolo, a versão e a versão do código de estado. Esta linha é seguida de vários cabeçalhos e do código da página propriamente dita (*entity body*).

```
HTTP Response Message

HTTP/1.1 200 OK
Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html
data data data data data ...
```

**Figura 4 - Exemplo de mensagem de resposta HTTP<sup>10</sup>**

Na figura 3, o pedido HTTP utiliza o método GET, que os *browsers* utilizam para efetuar o pedido de uma página, ou seja, indicam que pretendem ler uma página. Por sua vez, o método POST é utilizado para obter uma página cujo conteúdo depende de valores fornecidos pelo utilizador, com o objetivo de fazer a adição de valores a uma página *web*.

Considera-se também importante referir o tipo de ligação a utilizada. O HTTP especifica a ligação entre clientes e servidores pode ser de dois tipos: persistente e não-persistente (Kurose & Ross, 2007, p. 102). Segundo Boavida & Bernardes (2012), diz-se

<sup>9</sup> Fonte: Adaptado de Kurose & Ross (2007, p. 104)

<sup>10</sup> Fonte: Adaptado de Kurose & Ross (2007, p. 104)



que a ligação é não-persistente quando cada um dos objetos é transferido numa ligação própria, ou seja, para cada objeto pedido pelo cliente é necessário estabelecer uma sessão TCP<sup>11</sup> com o servidor, terminando a ligação assim que o cliente confirmar que recebeu o objeto. Já na ligação persistente, o servidor não termina a ligação TCP com o cliente, ficando a espera de novos pedidos de objetos que são transmitidos sobre a mesma ligação (Kurose & Ross, 2007, p. 103). O servidor termina a ligação quando decorrer um determinado tempo (*timeout*) sem que tenha recebido novos pedidos do mesmo cliente na ligação TCP estabelecida (Boavida & Bernardes, 2012).

HTTPS (*HyperText Transfer Protocol Secure*) é a combinação do protocolo HTTP com o SSL (*Secure Sockets Layers*) (Moreira, 2005). Tem por objetivo proteger o protocolo de rede que o servidor Web utiliza para efetuar a transferência de mensagens com a aplicação cliente. Com a implementação do SSL, a segurança é garantida através da cifra de toda a comunicação entre a aplicação cliente e o servidor Web (Kurose & Ross, 2007).

## MIME

O MIME (*Multipurpose Internet Mail Extensions*) é uma *norma* proposta pelos laboratórios *Bell Communications* em 1991 para alargar as possibilidades limitadas do correio eletrónico e nomeadamente permitir inserir documentos (imagens, sons, texto, etc.) numa mensagem (Freed & Borenstein, 1996). O MIME propõe descrever, graças a cabeçalhos, o tipo de conteúdo da mensagem e a codificação utilizada. Tem ainda a capacidade de atribuir ao serviço de mensagens as seguintes funcionalidades: a possibilidade de ter vários objetos (anexos) numa mesma mensagem, um comprimento de mensagem ilimitado, a utilização de jogos de caracteres (alfabetos) para além do ASCII, a utilização de texto enriquecido (formatação das mensagens, tipo de letras, cores, etc.) e anexo binários (executáveis, imagens, ficheiros áudio ou vídeos, etc.).

O tipo MIME é utilizado igualmente na Web para caracterizar os documentos transferidos pelo protocolo HTTP. Assim, quando uma transação entre um servidor web e um utilizador, o servidor web envia em primeiro lugar o tipo MIME do ficheiro enviado ao cliente, para que este último possa saber como apresentar o documento.

Kurose & Ross (2007) citam alguns exemplos de *MIME Types*:

- **Imagens:** *image/jpg*, *image/png*, *image/bmp*, *image/tiff*, entre outros;

---

<sup>11</sup> TCP, acrónimo para *Transmission Control Protocol*.

- **Estilo:** *text/css*;
- **Javascript:** *text/javascript*;
- **Documentos:** *application/x-excel*, *application/msword*, *application/xml*, etc.;
- **Áudio e vídeo:** *audio/mpeg*, *video/mpeg*, *audio/mid*, *video/avi*, entre outros

## 2.3. Aplicações Móvel

As aplicações móveis podem ser classificadas de acordo com vários critérios, o critério considerado relevante para esta dissertação refere-se ao tipo de infraestrutura utilizada para o seu desenvolvimento (Fling, 2009). As aplicações móveis nativas incluem-se neste grupo e são o tipo mais comum de aplicações. Estas são desenvolvidas e compiladas para dispositivos que executam uma determinada plataforma móvel, recorrendo a SDK<sup>12</sup> específicos e a linguagens de programação (Ribeiro, 2012).

Dada a natureza mais específica destas aplicações, torna-se possível utilizar um grande conjunto de funcionalidades dos dispositivos móveis para os quais são desenvolvidas, tal como as ligações de rede disponibilizadas pelo dispositivo móvel. É precisamente por esta razão que se torna na aplicação móvel mais adequada ao pretendido no desenvolvimento desta dissertação.

As aplicações nativas recorrem a *frameworks* para o seu desenvolvimento. Estas *frameworks* são infraestruturas designadas por API<sup>13</sup>. São normalmente lançadas para uma determinada plataforma, e correm no topo do sistema operativo, proporcionando funções que permitem aceder a serviços nucleares do mesmo. Dois exemplos destas infraestruturas incluem:

- **Cocoa Touch**, a infraestrutura que proporciona o acesso à API da Apple que permite desenvolver aplicações móveis nativas para os dispositivos multitoque da plataforma iPhone baseados no iOS. A programação de aplicações recorra à linguagem de programação Objective-C (Apple, 2012).
- **Android SDK**, trata-se de uma infraestrutura que permite programar aplicações nativas para dispositivos da plataforma Android. A utilização desta infraestrutura pode ser realizada recorrendo à linguagem de programação Java. Este SDK disponibiliza um depurador de código, um emulador e as bibliotecas que facilitam o acesso as funcionalidades do dispositivo Android (Android, 2012).

---

<sup>12</sup> SDK, acrónimo para *Software Development Kits*

<sup>13</sup> API é o acrónimo de *Application Program Interface*, refere-se a um conjunto de rotinas estabelecidas por um software para a utilização das suas funcionalidades por aplicações

As aplicações nativas proporcionam a melhor experiência para o utilizador do dispositivo móvel, podendo ser utilizadas *online* ou *offline*. A desvantagem destas aplicações deve-se ao facto de não poderem ser facilmente transportadas para outras plataformas móveis.

### 2.3.1. Plataformas

As plataformas são linguagens de programação que suportam o desenvolvimento e a execução de aplicações de *software* e serviços que acedem aos diferentes componentes que equipam o dispositivo móvel (Ribeiro, 2012). Existem vários tipos de plataformas móveis, que variam desde plataformas autorizadas com licenças, patenteadas (ou proprietárias) e de código aberto (*open source*). As plataformas autorizadas mediante a compra de licenças de utilização são vendidas aos fabricantes de dispositivos móveis para lhes permitir distribuí-las juntamente com o *hardware* que fabricam (Ribeiro, 2012).

#### Autorizadas com Licença

De acordo com Fling (2009), os exemplos mais comuns das plataformas autorizadas com licenças são: Java ME, Windows Phone, BREW e LiMo.

A plataforma Java ME (*Java Micro Edition*) é independente do sistema operativo e suporta uma versão reduzida da linguagem java, adaptada a dispositivos móveis com poucos recursos (Ribeiro, 2012).

A plataforma Windows Phone, baseia-se num sistema operativo, conhecido por Windows CE (*Windows Embedded Compact*), que foi especialmente desenvolvido e adaptado para dispositivos móveis e embebidos, combinada com um leque de aplicações básicas e baseada na API Windows da Microsoft que suporta aplicações que podem ser executadas em sistemas que variam desde 16bits até sistemas de 64bits.

A plataforma BREW (*Binary Runtime Environment for Wireless*), usada em dispositivos da Qualcomm, suportam as linguagens C, C++, Java e Flash Lite.

A plataforma LiMo é baseada em Linux e que foi criada pela Fundação LiMo. Apesar de o sistema operativo Linux ser um sistema *open source*, a plataforma LiMo foi desenvolvida para permitir o licenciamento por parte dos fabricantes de dispositivos móveis, e inclui SDK para o desenvolvimento de aplicações nativas baseadas em Java ou aplicações *Web* baseadas na infraestrutura de aplicações *Webkit browser Framework*.

## **Patenteadas**

No que diz respeito às plataformas patenteadas, trata-se de plataformas proprietárias que são concebidas pelos próprios fabricantes de dispositivos móveis para uso exclusivo nos respetivos dispositivos, não estando pois disponíveis para licenciamento por outros fabricantes. Os exemplos mais comuns incluem as plataformas Palm, Blackberry e iPhone (DevMedia, 2011).

A plataforma Palm foi desenvolvida inicialmente para PDA Palm Pilot com a designação de Palm OS tendo por base as linguagens de programação C e C++ e tendo posteriormente sido utilizada para *smartphone*. Mais recentemente, a Palm passou a basear os respetivos *smartphone* na plataforma Windows Mobile. Nos últimos tempos, a Palm voltou a alterar a plataforma de base para os seus dispositivos móveis, tendo introduzido a plataforma webOS (Ribeiro, 2012).

A plataforma Blackberry da RIM é uma plataforma patenteada desenvolvida com base na linguagem Java, destinada exclusivamente aos dispositivos Blackberry fabricados pela RIM.

A plataforma iPhone da Apple baseia-se numa versão reduzida do sistema operativo Mac OS X, designada por iOS, desenvolvida com base no sistema Unix e destinada exclusivamente a dispositivos móveis iPod, iPhone e iPad equipados com tecnologia *touchscreen* (Ribeiro, 2012).

## **Código Aberto**

Finalmente, as plataformas de código aberto encontram-se disponíveis gratuitamente, podendo ser descarregadas, editadas e modificadas por qualquer utilizador. Trata-se da categoria mais recente de plataformas móveis que tem conquistado muitos fabricantes de dispositivos e programadores de aplicações móveis. O exemplo desta categoria é a plataforma Android desenvolvida pela *Open Handset Alliance*, que é liderada pela Google. O objetivo principal desta organização é o desenvolvimento de uma plataforma *open source* inteiramente baseada na linguagem de programação Java (Ribeiro, 2012).

## **2.4. Dispositivos**

Tem-se verificado uma utilização crescente das tecnologias de transmissão sem fios para transferência de informação. Este enorme desenvolvimento deu origem ao

aparecimento de novos paradigmas de comunicação e de novos modelos de negócio (Darcey & Shane , 2012).

As comunicações móveis surgiram no mercado com o primeiro telefone comercial em 1983, tratava-se do Motorola dynaTAC *phone* 8000X (ver figura 5). Este aparelho, com cerca de 1kg, tinha autonomia de uma hora de conversa, oito horas em *standby*<sup>14</sup> e uma memória para trinta números (Darcey & Shane , 2012).



**Figura 5 -Telefone Motorola DynaTAC 8000X<sup>15</sup>**

Ao longo dos tempos, a evolução tecnológica deu primazia à criação de dispositivos menores com mais autonomia, mais memória e incremento de outras funcionalidades além das básicas chamadas de voz. São elas as mensagens de texto, *Multimedia Messaging Service*<sup>16</sup> (MMS), acesso à internet, comunicações de curto alcance sem fios, fotografia, jogos e as mais variadas aplicações. A tecnologia *wireless* elevou os patamares. Conceitos como conectividade e mobilidade criaram novos hábitos, relações e formas de trabalho. Esta nova tecnologia permite uma enorme mudança na forma de trabalhar, comunicar, divertir, estudar e qualquer outra atividade que preferimos fazer enquanto estamos em movimento (Viega, 2008).

Atualmente os dispositivos móveis são cada vez mais populares, dois exemplos que são amplamente utilizados pela nossa sociedade são os *smartphones* e os *tablets*. Os dispositivos móveis evoluíram em vários aspetos tecnológicos e ganharam novas designações. Estes dispositivos tornaram-se praticamente indispensáveis na vida pessoal e profissional de qualquer cidadão inserido numa sociedade moderna (Patrocínio, 2004).

---

<sup>14</sup> *Standby*, significa “em espera”. Termo utilizado em aparelhos eletrónicos para definir que o aparelho está ligado mas não está a ser utilizado, economizando assim energia.

<sup>15</sup> Fonte: <http://todayscampus.com/articles/load.aspx?art=1104> .

<sup>16</sup> MMS, significa serviço de mensagens multimédia. Tecnologia que permite a transmissão de texto, imagem, áudio e vídeo entre telemóveis ou entre telemóvel e correio eletrónico.

Os novos dispositivos móveis oferecem diversas funcionalidades além da tradicional capacidade de comunicação. Os *smartphones* possuem capacidade de comunicação com redes de computadores ou até mesmo com outros dispositivos móveis, demonstrando o grande sucesso da computação móvel. A competição por novas tecnologias e a iniciativa dos fabricantes mundiais evidenciou os dispositivos móveis em relação ao aperfeiçoamento e recursos (Costa, Duarte , & Filho, 2011).

Na sequência desta tendência, surgiu o conceito designado por BYOD (*Bring Your Own Device*) que é utilizado para denominar as situações em que as pessoas passam a utilizar os dispositivos móveis pessoais para realizarem o seu trabalho, em vez de utilizarem dispositivos que pertencem à organização empregadora. Isto significa que as pessoas trazem o seu próprio dispositivo para o local em que um determinado serviço é proporcionado (Ribeiro, 2012).

Normalmente, o surgimento de uma nova tendência, como é o caso do BYOD, trás consigo vantagens e desvantagens. No campo das vantagens importa considerar que a grande maioria dos dispositivos são pequenos, leves e simples de transportar. Têm ainda uma autonomia que dá para um dia de trabalho. Os elementos da organização preferem utilizar os seus próprios dispositivos, para que não tenham de se habituar a outros sistemas que não conhecem.

Já as desvantagens são muitas e variadas. É, por exemplo, difícil gerir o conteúdo e a configuração dos dispositivos. Não obstante do referido, é ainda mais difícil proteger e controlar o tráfego efetuado. Mas existem mais fatores a considerar, como as questões de cópia de documentos para o próprio dispositivo, que poderá dar origem a um potencial risco de segurança. Mesmo que não se tratem de documentos, os utilizadores poderão configurar as contas de correio eletrónico da organização nos seus próprios dispositivos, o que irá contribuir para o armazenamento local de muitas informações sensíveis.

Muitas aplicações comunicam frequentemente com a Internet. Estas comunicações são normalmente efetuadas de forma não segura. Utilizando ferramentas como o WireShark é possível aceder-se a todos os detalhes dessas comunicações (incluindo palavras-passe), permitindo a terceiros o acesso a informações confidenciais importantes.

Serão abordadas duas plataformas móveis cada vez mais procuradas no mercado pelos utilizadores devido às suas características de pequenas dimensões e elevada

capacidade tecnológica; os *smartphone* e os *tablets*. Serão ainda mencionados os dispositivos *ruggedized*<sup>17</sup>, direcionados para trabalhar em ambientes críticos.

#### 2.4.1. Smartphones

*SmartPhone*, que numa tradução livre de inglês significa telefone inteligente, é um telemóvel com funcionalidades avançadas que são alargadas por meio aplicações e executadas pelo seu sistema operativo (Pinheiro, 2011).

O *smartphone* engloba algumas das principais tecnologias de comunicação num mesmo local: internet, *Global Position System* (GPS), correio eletrónico, *Short Message Service* (SMS), entre outros. No que toca a conectividade o *smartphone* conta principalmente com *Bluetooth*, Wi-Fi e USB<sup>18</sup>. O *telefone* pode ser utilizado das mais diversas formas, como um controlo remoto ou uma bússola, por exemplo (Barros, 2012).

A principal vantagem de um *smartphone* é a sua polivalência. Ele serve para executar diversas tarefas que irão facilitar a vida do utilizador. O mercado dos *smartphones* cresceu muito nos últimos anos e, com isso, surgiram diversas opções, desde dispositivos com preços que se assemelham ao de telemóveis convencionais, até aos mais caros e mais sofisticados. O seu *design* moderno (ver figura 6), tamanho e peso reduzido, bem como o ecrã de alta resolução, a grande maioria com tecnologia *touchscreen*<sup>19</sup> (ecrã sensível ao toque), torna-o num dispositivo bastante atraente para o consumidor (Barros, 2012).



Figura 6 - SmartPhone Samsung Galaxy S<sup>20</sup>

<sup>17</sup> *Ruggedize*, termo inglês utilizado para mencionar uma versão mais robusta de alguma coisa, de modo a que esta tenha capacidade de suportar ambientes hostis, tem maior durabilidade que a versão normal.

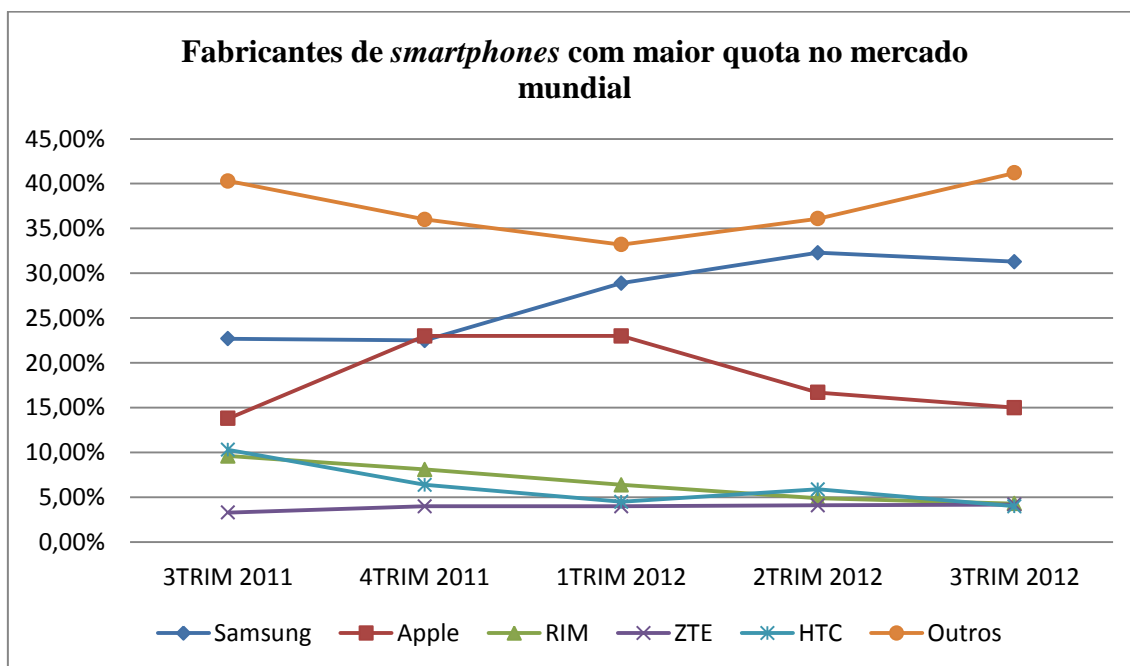
<sup>18</sup> USB, acrónimo de *Universal Serial Bus*. É o protocolo de comunicação que suporta transferência de dados entre computadores e periféricos.

<sup>19</sup> *Touchscreen* refere-se a um ecrã que é sensível ao toque de um dedo ou caneta.

<sup>20</sup> Fonte: <http://www.samsung.com>

Segundo dados disponibilizados pela *International Data Corporation* (IDC), as cinco principais marcas de *smartphones* no mercado atual são: Samsung, Apple, RIM, ZTE e HTC. A IDC é a empresa líder mundial na área de "*market intelligence*", serviços de consultoria e organização de eventos para os mercados das Tecnologias de Informação, Telecomunicações e Eletrônica de Consumo (IDC, 2012).

O gráfico 1 apresenta os dados disponibilizados pela IDC no que diz respeito aos *smartphones* mais vendidos no mercado a nível mundial, desde o terceiro trimestre de 2011 até ao terceiro trimestre de 2012. Os valores numéricos disponibilizados pela IDC, que deram origem ao gráfico encontram-se discriminados no Anexo A<sup>21</sup>. A tabela 1 apresenta a quantidade de *smartphones* vendidos a nível mundial, em milhões de unidades, no terceiro trimestre de 2011 e no respetivo trimestre de 2012.








**Gráfico 1 – Mercado mundial de vendas de *smartphone*, entre o terceiro trimestre de 2011 e o terceiro trimestre de 2012.**<sup>22</sup>

<sup>21</sup>Dados percentuais, disponibilizados pela IDC, sobre quota do mercado mundial dos principais fabricantes de *smartphones*

<sup>22</sup> Fonte: Adaptado de RESTIVO *et al*, outubro 2012.



Fabricante	3TRIM 2011	3 TRIM 2012	Comparação ano-a-ano
	(milhões de unidades)		
	28,1	56,3	100,4%
	17,1	26,9	57,3%
	11,8	7,7	-34,7%
	4,1	7,5	82,9%
	12,7	7,3	-42,5%
Outros	49,9	74	48,3%
TOTAL	123,7	179,7	45,3%

**Tabela 2 - Smartphones vendidos no mundo, em milhões de unidades.<sup>23</sup>**

Ao analisar estes dados, verificamos que a Samsung é a líder do mercado mundial de *smartphones*, a empresa coreana vendeu **56,3 milhões** de *smartphones* entre o início de julho e fim de setembro de 2012, número 100,4 por cento maior que no mesmo período do ano passado. A Apple também cresceu significativamente, com vendas de **26,9 milhões de iPhones**, um aumento de 57,3 por cento na comparação ano-a-ano. No geral, o mercado de *smartphones* cresceu 45,3 por cento no ano, alcançou **179,7 milhões de aparelhos vendidos**. A Nokia saiu dos primeiros lugares, a RIM ficou em terceiro lugar, vendeu 7,7 milhões de unidades, seguida da ZTE com 7,5 milhões e HTC com 7,3 milhões (IDC, 2012).

#### **2.4.2. Tablet**

Um *tablet* é um dispositivo pessoal que surge em forma de prancheta. Possui tecnologia *touchscreen* e tem o teclado no próprio ecrã, quando necessário. Este pode ser usado para acesso à internet, organização pessoal, visualização de fotos, vídeos, leitura de livros (*e-books*<sup>24</sup>), jornais e revista, entre outros. Um *tablet* não deve ser igualado a

<sup>23</sup> Fonte: Adaptado de RESTIVO *et al*, outubro 2012.

<sup>24</sup> *e-book* é uma abreviação do termo *eletronic book*, que significa livro em formato digital.

um computador completo ou um *smartphone*, embora possua funcionalidades de ambos (Couto, 2012). Os primeiros dispositivos deste tipo que chegaram ao mercado foram chamados *Tablet PCs*. Possuíam uma caneta especial que acionava as funcionalidades do dispositivo através do toque. Utilizavam os mesmos sistemas operativos presentes nos computadores convencionais. Após o lançamento do iPad pela Apple, em 2010, a popularidade deste dispositivo aumentou e ganhou grande aceitação por parte dos consumidores (ver figura 7). Esta nova versão de *tablet* utiliza um ecrã LCD<sup>25</sup> sensível ao toque dos dedos, sendo dispensável o uso de caneta especial. O seu sistema operativo também se distingue dos sistemas operativos existentes nos computadores comuns (Couto, 2012).



Figura 7- iPad da Apple<sup>26</sup>

Surgem então outros fabricantes a desenvolver *tablets* com recursos semelhantes ao iPad e com diferentes sistemas operativos como é o caso do *Android*. Inicialmente o sistema operativo *Android* ainda não oferecia otimização para correr em *tablets*, pois foi desenvolvido para funcionar apenas em *Smartphones*. Em 2011 foi lançada uma nova versão do sistema operativo da Google para dar resposta à necessidade de otimização para *tablets*: a versão 3.0 apelidada de *Honeycomb*. Com isto, surge no mercado uma vasta gama de *tablets* de diversas marcas com o sistema operativo *Android* e capaz de competir com o iPad (Couto, 2012).

Atualmente vários fabricantes adotaram uma gama de *tablets* considerados híbridos. Esta denominação deve-se a uma mistura de *tablet* com *notebook*<sup>27</sup> normal. Estes modelos têm um *design* que permite utilizar a tecnologia *touchscreen* e também o teclado. Para

---

<sup>25</sup> Acrónimo para *liquid crystal display*, um display de cristal líquido.

<sup>26</sup> Fonte: <http://www.apple.com>





<sup>27</sup> *Notebook* refere-se a um computador portátil, leve, designado para ser transportado e utilizado em diversos locais com facilidade.

alterar entre um estado e outro, em alguns casos, basta girar o ecrã, deixando-o sobre o teclado. Noutros modelos o *tablet* traz um teclado extra que pode ser acoplado quando é necessário escrever documentos, como é ilustrado na figura8.



**Figura 8 - Tablet Híbrido Lenovo IdeaPad<sup>28</sup>**

Segundo dados disponibilizados pela IDC, as quatro principais marcas de *tablets* no mercado atual são: Apple, Samsung, ASUS e Acer (IDC, 2012). A tabela 2 apresenta a quantidade de *tablets* vendidos a nível mundial, em milhões de unidades, no segundo trimestre de 2011 e no respetivo trimestre de 2012.

Fabricante	2TRIM 2011	2TRIM 2012	Comparaçã o ano-a-ano
	(milhões de unidades)		
	9,248	17,042	84,3%
	1,099	2,391	117,6%
	0,397	0,855	115,4%
	0,629	0,385	-38,7%
Outros	3,668	3,067	-16,4%

**Tabela 3 - Tablets vendidos no mundo, em milhões de unidades<sup>29</sup>**

<sup>28</sup> Fonte: <http://www.lenovo.com>

<sup>29</sup> Fonte: Adaptado de MAINELLI, agosto 2012.

Ao analisar estes dados, verifica-se que a Apple é a líder do mercado mundial de *tablets*, a empresa vendeu **17 milhões de tablets** entre o início de abril e fim de junho de 2012. Contudo o maior crescimento foi da Samsung que vendeu 117,6 vezes mais que no mesmo período do ano passado. A Amazon.com surgiu recentemente mas com grande procura por parte dos consumidores, com 1,252 milhões de unidades vendidas, seguido da Asus com 855 mil unidades e que também teve um crescimento acentuado. A Acer perdeu algum terreno, vendeu 386 mil unidades no mesmo trimestre. No geral, o mercado de *tablets* cresceu 66,1 por cento no ano, alcançou **24,994 milhões de aparelhos vendidos** (IDC, 2012).

### 2.4.3. *Ruggedized*

O termo *Ruggedized* significa robustez. Os dispositivos móveis com esta designação possuem algum nível de proteção contra ambientes hostis. Pode ser um equipamento que inclui um invólucro de borracha para proteção contra quedas, ou pode possuir um corpo mais resistente proporcionando maior durabilidade e proteção contra choques, bem como resistência à água e poeiras. O principal público-alvo que exige este grau de robustez nos seus dispositivos são, por exemplo, as Forças Armadas.

Segundo um artigo lançado pela revista PC GUIA, a Rússia está a desenvolver o seu próprio *tablet* e que se destina às suas forças armadas, baseado no sistema operativo móvel da Google. O dispositivo será denominado de *Russian Mobile Operating System* (RoMOS), o seu principal consumidor será o Ministério da Defesa russo. A versão militar do *tablet* estará preparada para ser utilizada em condições adversas. O *tablet* deverá ser lançado na Rússia até ao final do ano 2012 (Racoma, 2012).

Existem muitos *tablets* e *smartphones* robustos no mercado que se destinam a profissionais que trabalham em locais adversos. A Panasonic e a Casio possuem estes dispositivos robustos no mercado que usam o sistema operativo Android 4.0 (*Ice Cream Sandwich*) e apresentam um invólucro em borracha que os protege se caírem de uma altura de um metro, sendo ainda resistentes à água e à poeira (ver figura 9).



Figura 9 - Tablet Panasonic ToughPad<sup>30</sup>

## 2.5. Conclusões

Este capítulo efetua uma abordagem ao estado da arte tecnológico da primeira camada da arquitetura protótipo, descrevendo assim os seus componentes, que permitirá escolher os elementos que vão definir a arquitetura alvo.

Os protocolos podem ser entendidos como regras que regem as trocas de informação nas redes de computadores. Fez-se a revisão das características do protocolo HTTP para aplicações Web.

Nas aplicações móveis definiu-se que a que mais se adequa ao desenvolvimento do projeto seria uma aplicação móvel nativa. Foram discriminadas algumas *frameworks* para o desenvolvimento de aplicações móveis. Foram vistas algumas das plataformas mais importantes no mercado dos diferentes tipos: autorizadas com licença, patenteadas e de código aberto.

Ao nível dos dispositivos, identificou-se a tendência de computação móvel e de BYOD. Foram abordados os dispositivos móveis com maior popularidade no mercado. Verificou-se ainda que existe no mercado uma vasta gama de *tablets* e *smartphones* robustos (*ruggedized*) de várias marcas. Estes têm maior proteção e resistência que os *tablets* normais, são ideais para profissionais que trabalham em situações hostis, como é o caso dos militares, especificamente, que trabalham em unidades navais.

---

<sup>30</sup> Fonte: <http://www.panasonic.com>



## CAPÍTULO III SERVIÇOS TÉCNICOS

### 3.1. Servidores Web

Um servidor Web pode ter duas vertentes: um computador onde são hospedados as páginas web e um programa que corre no mesmo computador (*host*). Assim, a terminologia servidor web refere-se ao *hardware* e *software* (Pinheiro J. , 2010).

Servidor web (programa) é um software que é executado no computador servidor responsável pela hospedagem dos *websites*. Com objetivo de disponibilizar as páginas web, o que significa que responde a pedidos dos clientes (*browsers*) (Pereira & Poupá, 2011). Cada lado exige também um programa especializado para a troca de dados. No caso do cliente, é usado por exemplo um *browser*, como o *Internet Explorer* da Microsoft ou *Chrome* da Google (Kurose & Ross, 2007)

No lado do servidor existem várias opções de *software* disponível. A tarefa destes baseia-se em negociar a transferência de dados entre clientes e servidores via HTTP, o protocolo de comunicações da Web. O *software* depende do sistema operativo escolhido para o servidor. Quando um utilizador, no *browser*, efetua o pedido de um documento (indicado por um URL<sup>31</sup>) a um servidor, obedecendo os seguintes passos:

O servidor, ao receber o pedido (URL) do utilizador, executa a página web correspondente ao URL. O conteúdo produzido pela página, normalmente um documento no formato HTML, é uma combinação de marcas (*tags*) HTML (incluídos explicitamente) e o resultado de algum processamento (por exemplo, algoritmo Java e/ou acesso a um SGBD);

O utilizador recebe o conteúdo criado pelo servidor exibe-o através do seu *browser*.

Serão analisados alguns servidores Web existentes no mercado (Pereira & Poupá, 2011).

#### 3.1.1. IIS

O *Internet Information Services* (IIS) é um servidor web criado pela Microsoft para seus sistemas operativos. A primeira versão foi introduzida com o WindowsNT Server versão 4, e passou por várias atualizações. Uma de suas características mais utilizadas é a geração de páginas HTML dinâmicas, que ao contrário de outros servidores web, usa tecnologia proprietária, o ASP.NET, que será analisado no ponto

---

<sup>31</sup> Acrónimo para *Uniform Resource Locator*.

3.2.3., mas também pode utilizar outras tecnologias com adição de módulos de terceiros (Miranda, 2010).

Para utilizar essa ferramenta é necessário adquirir licença de uso, onde cada instalação ou versão é paga, que é considerado uma desvantagem comparativamente a outros tipos de servidores (Microsoft, 2012).

### **3.1.2. Apache**

O servidor Apache é o mais bem sucedido dos servidores de código aberto. Foi criado em 1995 por Rob McCool (Almeida, 2011).

Este servidor é compatível com os protocolos HTTP/HTTPS, sendo que as suas funcionalidades são mantidas através de uma estrutura de módulos, permitindo inclusive que o utilizador escreva os seus próprios módulos (Souza, 2011).

Num servidor Web, o aspeto da segurança é muito importante. Deste modo, para garantir a integridade nas trocas de mensagens HTTP, o Apache dispõe de um módulo chamado *mod\_ssl*, o qual adiciona a capacidade do servidor atender pedidos utilizando o protocolo HTTPS. O Apache é configurado de um modo simples e visível para o administrador do sistema. Utiliza um conjunto de ficheiro de texto (em claro), nos quais são inseridas as opções de funcionamento do utilizador. Tem a desvantagem que, não dispondo de uma ferramenta de configuração do tipo visual (janela de interação com o utilizador), é necessário que o administrador do serviço tenha conhecimento de todos os parâmetros de configuração que são necessários. A grande vantagem deste sistema de configuração é que qualquer *software* de edição de texto pode ser usado na configuração do servidor. (Souza, 2011).

### **3.1.3. Apache Tomcat**

O servidor Tomcat foi desenvolvido pela fundação Apache e permite a execução de aplicações para a web, e tal como o Apache, é um *software* de código aberto. Tem como principal característica técnica a utilização da linguagem de programação Java, mais especificamente nas tecnologias de *Servlets* e de JSP<sup>32</sup> (Pereira & Poupa, 2011). Esta abordagem concorre com a usada pela Microsoft com o ASP.NET (baseada na linguagem Visual Basic e C#). O desenvolvimento de uma típica aplicação web a ser executada pelo Tomcat implica o domínio das seguintes linguagens: HTML, XML e a tecnologia JSP. O servidor Tomcat tem a habilidade de converter automaticamente qualquer página JSP em

---

<sup>32</sup> Acrónimo para *Java Server Pages*



um *servlet* equivalente. Por outro, o Tomcat é capaz de criar código fonte Java a partir de um documento HTML (DevMedia, 2011).

Segundo a DevMedia (2011), no ponto de vista operacional, a principal finalidade das tecnologias de JSP é permitir a criação dinâmica de conteúdos.

## 3.2. Tecnologias

### 3.2.1. Introdução

Em março de 1989 “nasceu” a *Web*, quando Tim Berners-Lee do CERN<sup>33</sup> propôs o protocolo HTTP como meio de desenvolvimento de um sistema hipermédias distribuído que permitisse o acesso e a partilha de informação de investigação, incluindo documentos, relatórios e imagens, entre os servidores de vários centros de investigação. Esta forma de partilhar informação foi tão bem-sucedida que conduziu ao crescimento exponencial da *Web* que se assiste atualmente (Ribeiro, 2012).

O desenvolvimento *Web* é o futuro, há milhões de utilizadores conectados na grande rede de computadores. Muitos destes, com diversos interesses e sempre com algum objetivo: criar novos amigos, comprar produtos, jogar e tantas outras tarefas que a *Web* pode proporcionar através da Internet (Viega, 2008)

Para o desenvolvedor, o que realmente importa são as tecnologias que estão por trás deste mundo fantástico que é a *Web*. Existem muitas linguagens de programação *Web* (também conhecidas de linguagem de *scripting*), cada uma com seus prós e contras. Mas, atualmente há três linguagens (na ótica do servidor) que são muito utilizadas na construção de *sites*: JSP, ASP e PHP (Pereira & Poupa, 2011).

O tipo de linguagem a utilizar na construção de uma aplicação *Web* está dividido em duas partes: linguagem na ótica do servidor, conhecidas como *server-side*; as linguagens na ótica do cliente, que são conhecidas como *cliente-side* (Pereira & Poupa, 2011).

---

<sup>33</sup> CERN, Organização Europeia para a Pesquisa Nuclear, antigo acrónimo para *Conseil Européen pour la Recherche Nucléaire*

### 3.2.2. JSP

*Java Server Pages* (JSP) - é uma tecnologia utilizada no desenvolvimento de aplicações para *Web*. Por ser baseada na linguagem de programação Java, tem a vantagem da portabilidade de plataforma, que permite a sua execução em diversos sistemas operativos, como o Windows, Unix e Linux. Esta tecnologia produz aplicações que acedam a base de dados, manipulem arquivos em formato de texto, obtenham informações a partir de formulários e guardem informações sobre o utilizador e sobre o servidor. Uma página é criada com a tecnologia JSP, após instalada num servidor de aplicação compatível com a tecnologia Java EE<sup>34</sup> (Hackenhaar, Zanella, & Cardoso, 2010).

O JSP foi implementado para apoiar a criação de páginas HTML e XML. É uma linguagem que facilita a combinação de dados de modelo estático com conteúdo dinâmico (Hackenhaar, *et al.*, 2010).

Em relação a arquitetura, o JSP pode ser visto como uma abstração de alto nível dos *java servlets*<sup>35</sup>. Páginas JSP são carregadas dentro do servidor e processado a partir de uma estrutura especial do servidor Java, chamada *Java EE Web Application*, muitas vezes como um arquivo *.war* *.ear*. O JSP permite que o código Java e certas ações pré-definidas, sejam intercalados com conteúdo de marcação estático, assim, a página é compilada e executada no servidor para posteriormente ser entregue no formato HTML ou XML (Almeida, 2011).

A sintaxe do JSP é uma mistura de dois tipos básicos de conteúdo: *scriptlet elements* e *markup*. *Markup* é tipicamente um padrão HTML ou XML, enquanto os elementos *scriptlet* são blocos de código Java, que podem ser unidos com o tipo de marcação (Diogo & Graça, 2011).

#### Vantagens

Segundo Diogo & Graça (2011), destacam-se as seguintes vantagens:

- É uma linguagem orientada para objetos, deste modo, dá a possibilidade de criar programas em módulos e usar os mesmos novamente;

---

<sup>34</sup> Acrónimo para *Java Platform, Enterprise Edition* é uma plataforma de programação para servidores na linguagem de programação Java;

<sup>35</sup> *Servlet*, módulo de extensão da funcionalidade de um servidor Web, através de aplicações implementadas em Java, também pode ser considerado como uma classe na linguagem de programação Java.

- É uma linguagem segura, pois permite fazer uma pré-simulação da execução para verificar se existem efeitos negativos do código no computador;
- É uma tecnologia com alto grau de confiabilidade, destacando-se na verificação de erros na altura da compilação do programa;
- É independente de plataforma. Esta linguagem tem a capacidade de passar facilmente de um sistema operativo para outro sem erros de compatibilidade, ou seja, trabalhar em Windows ou em Linux é exatamente igual;
- Tem um conjunto de classes que implementam mecanismos de acesso remoto, as designadas classes de redes.

### 3.2.3. ASP.NET

Uma das linguagens utilizadas frequentemente no mercado de desenvolvimento de páginas *Web* é a ASP.NET. O ASP Clássico, ou simplesmente ASP, era uma linguagem de *script* que foi feita para ser executada em servidores Windows com IIS (Almeida, 2011).

O ASP.NET é uma tecnologia desenvolvida pela Microsoft, que veio suceder a tecnologia ASP. O ASP é uma linguagem com diversos recursos, possui uma sintaxe muito parecida com a do *Visual Basic* (VB), daí a grande aceitação no mercado. A produtividade a linguagem apresentou algumas debilidades e deixava muito a desejar. Construir uma página em ASP era trabalhoso, e partindo do princípio que tudo que dá trabalho tende a criar maiores custos para as empresas, a Microsoft, começou a perder utilizadores para o seu principal concorrente, o PHP uma linguagem orientada para objetos com amplos recursos (Almeida, 2011).

O ASP.NET é a mais nova geração de linguagem para desenvolvimento *Web*, da Microsoft. Juntamente com a plataforma .NET foi desenvolvida uma tecnologia que dá suporte a programação *Web* baseada em eventos e suporte a orientação a objetos. São muitos os benefícios que o ASP.NET trouxe, dos quais se destacam: uma vasta biblioteca de classes, pois o ASP.NET utiliza o .NET Framework; diversos recursos de OOP (Programação Orientada a Objetos) (Pereira & Poupa, 2011).

Além dos recursos supracitados, há ainda o *code-behind*<sup>36</sup> que separa o código do projeto do *site*. Este recurso dá a possibilidade de escolher a linguagem para programar a página: C# ou VB.NET (Pereira & Poupá, 2011, p. 182).

Através do ASP.NET é possível criar páginas *Web* completas, interativas utilizando os grande recursos da plataforma .NET (Almeida, 2011).

É importante frisar que uma aplicação para *web* desenvolvida em ASP.NET pode reutilizar código de qualquer outro projeto escrito para a plataforma .NET, mesmo que em linguagem diferente. Uma página ASP.NET escrita em VB.NET pode chamar componentes escritos em C# ou *Web Services* escritos em C++, por exemplo. Ao contrário da tecnologia ASP, as aplicações ASP.NET são compiladas antes da execução, aumentando o desempenho (Almeida, 2011).

Em rigor, segundo Pereira & Poupá (2011), o ASP.NET não é uma linguagem de programação, mas sim uma tecnologia de servidor que pode ser implementada através de uma linguagem de programação.

## **Vantagens**

De acordo com Pereira & Poupá (2011), O ASP.NET apresenta as seguintes vantagens:

- Tecnologia particularmente adequada à integração de *sites* com base de dados, em principalmente em base de dados da Microsoft.
- Favorece a separação entre o desenho de um determinado documento e a programação subjacente, o que permite que o *designer* e o programador trabalhem de forma independente.

## **Desvantagens**

- Como não é uma tecnologia de código aberto, os programas que permitem a sua implementação têm de ser pagos/comprados;
- Opera apenas com os produtos da Microsoft, no que concerne aos sistemas de gestão de base de dados e do servidor *Web*;
- Visto que esta tecnologia é dispendiosa, apenas grandes empresas, estudantes com acesso à plataforma através das universidades e afins têm possibilidade de a utilizar.

---

<sup>36</sup> *Code behind*, Técnica de separação do código em dois ficheiros: um, em HTML e ASP.NET para interface gráfica; outro, em VB.NET ou C# para o código que confere as funcionalidades a essa interface.

### 3.2.4. PHP

A sigla PHP significa Pré-Processador de Hipertexto (em inglês PHP *Hypertext Preprocessor*), é uma linguagem de *scripting* com o objetivo de gerar documentos HTML. O PHP é interpretado e executado no lado do servidor, permitindo o acesso a base de dados (Pereira & Poupa, 2011).

É a linguagem mais utilizada para criar páginas *web* na internet, dado ao facto de ser de código aberto. Além de ser código aberto, também é compatível em qualquer das plataformas que se seguem: Unix, Linux, Windows, Solaris, HPUX, sem esquecer que também é compatível com vários servidores *Web*, que são abordados na secção 1.4 (Almeida, 2011).

#### Vantagens

De acordo com Almeida (2011) e Pereira & Poupa (2011), as vantagens que do PHP são as seguintes:

- Permite o desenvolvimento rápido de aplicações;
- As aplicações são mais estáveis e não dependem só navegador utilizado pelo cliente;
- Liberdade de escolha da plataforma servidora (servidor *Web*);
- Tem ligações aos sistemas de gestão de base de dados mais utilizados;
- Tem uma grande comunidade de programadores e soluções já implementadas.

#### Desvantagens

Segundo Pereira & Poupa (2011), o PHP apresenta os seguintes inconvenientes:

- Não é uma linguagem nativamente orientada para objetos, apesar de apresentar melhorias em cada versão da mesma linguagem

### 3.3. Linguagens

As linguagens de marcas consistem num documento composto por três componentes, claramente distintos e separados, são eles: conteúdo, estrutura e estilo. O conteúdo é a informação propriamente dita, a estrutura define a organização da informação no documento, e o estilo define o aspeto de apresentação das informações ao utilizador. Tal distinção promove uma simplificação, pois o autor não tem de se preocupar *a priori* com o aspeto da informação, pelo que pode dedicar-se exclusivamente ao conteúdo e à estrutura de apresentação do documento. Desta forma, o texto mantém-se "limpo", eliminando

códigos que não dizem respeito ao conteúdo da informação, o que facilita a compreensão pelo homem. A linguagem XML e a linguagem HTML, bem como a sua evolução serão analisadas de seguida.

### 3.3.1. XML

O *eXtensible Markup Language* (XML) é uma linguagem para estruturação de documentos (Dykes & Tittel , 2005). A sua simplicidade e as suas características são atrativas quer para os programadores, quer para os utilizadores. Esta linguagem permite ao utilizador criar a sua própria linguagem de marcas (*tags*). Desta forma, a especificação de uma aplicação será um documento anotado com um conjunto de marcas apropriadas para definir o conteúdo da sua base de dados (Heitlinger, 2001).

O XML foi definido em 1998 pelo W3C, para representação de dados e contém um conjunto de regras de sintaxe fixas e obrigatórias. Essas regras determinam o formato que as *tags* devem ter, os caracteres permitidos e os seus significados, assim como o formato do documento em geral (W3C, 2005). A especificação XML não define quais as *tags*, ou seja, os elementos a usar, nem o que significam. O vocabulário é criado pelo utilizador do XML, conforme as suas necessidades. As principais características do XML são: a capacidade de separação do conteúdo e da formatação; o facto de se legível por humanos e por máquinas; a possibilidade de criação de *tags* sem limitação; a criação de documentos de validação de estrutura; e a capacidade de interligar bases de dados distintas (W3C, 2005).

De acordo com Dykes & Tittel (2005), um documento XML é composto basicamente por três elementos:

1. Conteúdo – Informação a ser disponibilizada;
2. Estrutura – Definida por um *Document Type Definition* (DTD) ou *Schema* que contém as regras de formatação;
3. Apresentação – A forma como os dados serão exibidos. Para isso utiliza-se *Cascading Style Sheets* (CSS) ou *eXtensible Style Language Transformation* (XSLT).

Assim, os arquivos de formato XML tornam-se auto descritivos, o que facilita a leitura, criação e extensão. Conta ainda com o suporte de empresas líderes no mercado de *software* e que determinou o seu rápido avanço, é o caso da Microsoft, Oracle, entre outras. Em pouco tempo, o XML tornou-se no formato universal para a partilha de dados entre aplicações. A facilidade declarações precisas, não só dos conteúdos de um documento,

como também dos elementos convenientes à estruturação desses conteúdos (Dykes & Tittel , 2005).

São muitas as vantagens do formato XML, a sua simplicidade, capacidade de separação entre conteúdo e formatação, capacidade de criar um número ilimitado de marcadores e facilidade na conversão de dados. Além disso é um formato de documentos aberto, extensível e não proprietário, ideal para a troca de dados entre múltiplas aplicações e plataformas (Dykes & Tittel , 2005).

### **3.3.2. HTML**

As páginas web constituem o recurso mais pedido aos servidores HTTP e são desenvolvidos através da linguagem HTML. Uma página HTML consiste num ficheiro de texto escrito na linguagem HTML, um formato de descrição da estrutura para documentos de texto que utiliza marcas para especificar a formatação e a estrutura do texto (Ribeiro, 2012). As funcionalidades básicas da linguagem HTML permitem a criação da estrutura e da formatação de páginas web estáticas com uma interatividade limitada (Ribeiro, 2012).

A ampliação dos elementos interativos pode realizar-se recorrendo ao desenvolvimento de *scripts* ou objetos que são inseridos nas páginas HTML e executados no lado do cliente, ou seja, pelo *browser*. Sendo uma linguagem de marcas, a HTML baseia-se na inserção de identificadores especiais no texto de um documento, de modo a delimitar porções de texto para as quais se pretende especificar um conjunto de propriedades, tais como a fonte do texto, a sua dimensão, o seu estilo e a sua cor. A utilização de marcas permite, assim, separar o processo de criação do conteúdo do processo utilizado para apresentar esse conteúdo (Coelho P. , 2004).

A linguagem HTML possui algumas limitações ao nível da interatividade, pois foi criada originalmente apenas com o objetivo de implementar hipertextos (Coelho P. , 2004).

### **3.3.3. HTML4**

As limitações gráficas da linguagem HTML original e o número reduzido de marcas que esta incluía deram origem a uma nova versão da linguagem, a versão HTML4 que foi emitida pelo W3C em 1997. Esta versão inclui um conjunto de melhorias que incluem uma separação entre as marcas que se destinam a especificar a estrutura da página Web e as marcas que se destinam à formatação dos conteúdos, bem como uma definição mais

concreta de elementos complexos, tais como as tabelas e as *frames*. A grande funcionalidade inovadora que foi introduzida com a linguagem HTML4 é relativa às folhas de estilos CSS. As folhas de estilo permitem definir o aspeto das páginas HTML de um modo completamente independente das marcas tradicionais e do conteúdo (Coelho, 2004).

Apesar da linguagem HTML4 apresentar um conjunto melhorado de características, quando comparado com versões anteriores, continua a existir um conjunto fixo de marcas. Esta limitação foi ultrapassada com o surgimento da linguagem XML em 1998. O XML, para além de reforçar a independência total entre o conteúdo e a apresentação de um documento, proporciona a sintaxe normalizada para criar conjuntos de marcas personalizadas. No entanto, ao contrário da linguagem HTML, a linguagem XML é uma linguagem que serve para definir outras linguagens (Ribeiro, 2012).

Recentemente, a linguagem HTML4 foi reformulada como uma aplicação da linguagem XML, dando origem à linguagem XHTML (*eXtensible HyperText Markup Language*), que combina o potencial expressivo da linguagem XML com o potencial de apresentação da linguagem HTML. Esta combinação resulta numa linguagem que permite visualizar as páginas Web em navegadores destinados a uma grande variedade de plataformas e dispositivos (Ribeiro, 2012).

### **3.3.4. AJAX**

Garret (2005) designou por “Ajax”, abreviatura de *Assynchronous JavaScript and XML*, a um conjunto de várias tecnologias, onde cada uma proporciona funcionalidades autónomas, alargando assim a quantidade de possibilidades disponíveis para o desenvolvimento de aplicações Web. O Ajax possui as seguintes características: criação de páginas Web baseada em normas internacionais, mediante a utilização de XHTML e CSS; apresentação dinâmica e interação com o DOM (*Document Object Model*); transferência e manipulação de dados baseados em XML e XSLT; transferências assíncronas baseadas no comando *XMLHttpRequest*; utilização da linguagem *JavaScript* para interligar todos estes componentes (Ribeiro, 2012).

As aplicações Ajax vieram substituir as transferências entre os clientes e os servidores do tipo pedido e resposta, através da introdução de um componente intermedio, designador por motor Ajax, que se situa entre o cliente e o servidor. Assim passou a ser possível atualizar partes de uma página Web sem ser necessário voltar a transferir a página inteira a partir do servidor. As aplicações Ajax baseiam-se num modelo de programação



baseado em eventos. Estes eventos correspondem às ações do cliente, que corresponde a chamadas de funções associadas a elementos da página Web. A interatividade é conseguida através da utilização de formulários e botões. O uso do modelo DOM permite associar ações aos elementos da interface da página Web e também extrair dados de documentos XML fornecidos pelo servidor (Ribeiro, 2012).

A Google foi a primeira organização a investir no desenvolvimento de aplicações Ajax, conforme se pode verificar em todas as aplicações que tem vindo a desenvolver para a Web, tais como o Gmail<sup>37</sup> e o Google Maps<sup>38</sup>. A Ajax marcou um desenvolvimento importante no que diz respeito à interação com as aplicações Web, e prestou um contributo importante para as novas funcionalidades que vieram a ser implementadas na linguagem HTML5 (Ribeiro, 2012).

### 3.3.5. HTML5

O WHATWG (*Web Hypertext Application Technology Working Group*) é uma comunidade constituída por entidades interessadas em fazer evoluir a norma HTML, e no desenvolvimento de API (Interfaces para a Programação de Aplicações) necessárias para a criação de aplicações Web (Abreu, 2011). Este grupo foi fundado em 2004 por indivíduos provenientes de diferentes empresas ligadas ao desenvolvimento para a Internet, são elas a Apple, a Mozilla Foundation e a Opera Software. Estas empresas encontravam-se apreensivas pelo rumo que a nova norma W3C, o XHTML, estava a tomar. O objetivo da WHATWG é a normalização da linguagem HTML, a qual engloba a programação, o alojamento de conteúdos na Internet, os eventos para a troca de mensagens e API para a troca de informação (Abreu, 2011). Em 2006, o W3C transformou o projeto que estava a ser desenvolvido pelo WHATWG na norma HTML5. A linguagem HTML5 surge como uma evolução da linguagem HTML e não como uma nova linguagem. A norma HTML5 pretende reduzir a necessidade de recorrer à utilização de *plugins* externos, trazendo algumas funcionalidades para os navegadores Web. Esta norma tenta criar um meio para uniformizar plataformas e dispositivos (Abreu, 2011).

Um dos principais objetivos desta nova norma é a de facilitar a manipulação dos elementos, possibilitando uma maior flexibilidade no que diz respeito à sua formatação

---

<sup>37</sup> O Gmail (também GMail, Google Mail) é um serviço gratuito de webmail criado pela Google em 2004. Atualmente novos cadastros são abertos ao público e podem ser feitos diretamente

<sup>38</sup> Google Maps é um serviço de pesquisa e visualização de mapas e imagens de satélite da Terra gratuito na web fornecido e desenvolvido pela Google

visual, tornando a experiência do utilizador final a mais transparente possível. Uma melhoria em relação a versões anteriores consiste na existência de API desenvolvidas de raiz que podem ser manipuladas através de CSS e *JavaScript* (Ribeiro, 2012).

Nesta norma são descritas novas marcas e respetivas características, bem como a forma correta de gestão de erros, mas reutiliza API externas que são, em si próprias, normas diferentes. Estas novas marcas estruturais vieram permitir definir e organizar os conteúdos de uma forma mais estruturada (Coelho P. , 2005).

A linguagem HTML5 permite o desenvolvimento de aplicações Web completas sem ser necessário o recurso a plataformas de desenvolvimento externas. Isto deve-se ao conjunto de novas marcas introduzidas que possibilitam a criação de aplicações destinadas a ser reproduzidas de modo nativo, diretamente no *browser*. Algumas dessas marcas focam a interatividade., ao passo que outras proporcionam um controlo melhorado sobre os formulários ou métodos adicionais para a formatação de conteúdos multimédia (Ribeiro, 2012).

Na norma HTML5 a distinção entre a especificação da linguagem HTML e a API do DOM foi eliminada, uma vez que o DOM passou a fazer parte da especificação da linguagem HTML5 (Abreu, 2011). A intenção é de permitir a interoperabilidade entre os dispositivos e o *browser*. Foram adicionados vários métodos e eventos para melhorarem a gama de funcionalidades oferecidas pela linguagem, onde se destaca o processamento de formulários e a manipulação de elementos interativos. Para além da inclusão das definições do DOM, a norma HTML5 passou também a incluir novas API, acessíveis através de *scripting*, como o propósito de transformar o *browser* numa plataforma nativa de programação, reduzindo a necessidade de se recorrer à utilização de outros programas de edição (Ribeiro, 2012).

A linguagem HTML5 aumenta a ênfase na distinção entre a semântica e a estrutura das páginas HTML pelo que o modelo de conteúdos sofreu uma reformulação, passando a existir novos modelos de conteúdos que iniciam a partilha de formatos de conteúdos, permitindo aos programadores o desenvolvimento de aplicações Web com estruturas mais sofisticadas, ao mesmo tempo que desenvolvem código mais simples (Ribeiro, 2012).

### **3.4. Serviços de Informação e Integração**

Os sistemas operativos (SO) móveis são serviços nucleares utilizados para comunicar com o *hardware* e permitem que outros programas, chamadas aplicações, possam ser executados no dispositivo (Darcey & Shane , 2012). São projetados

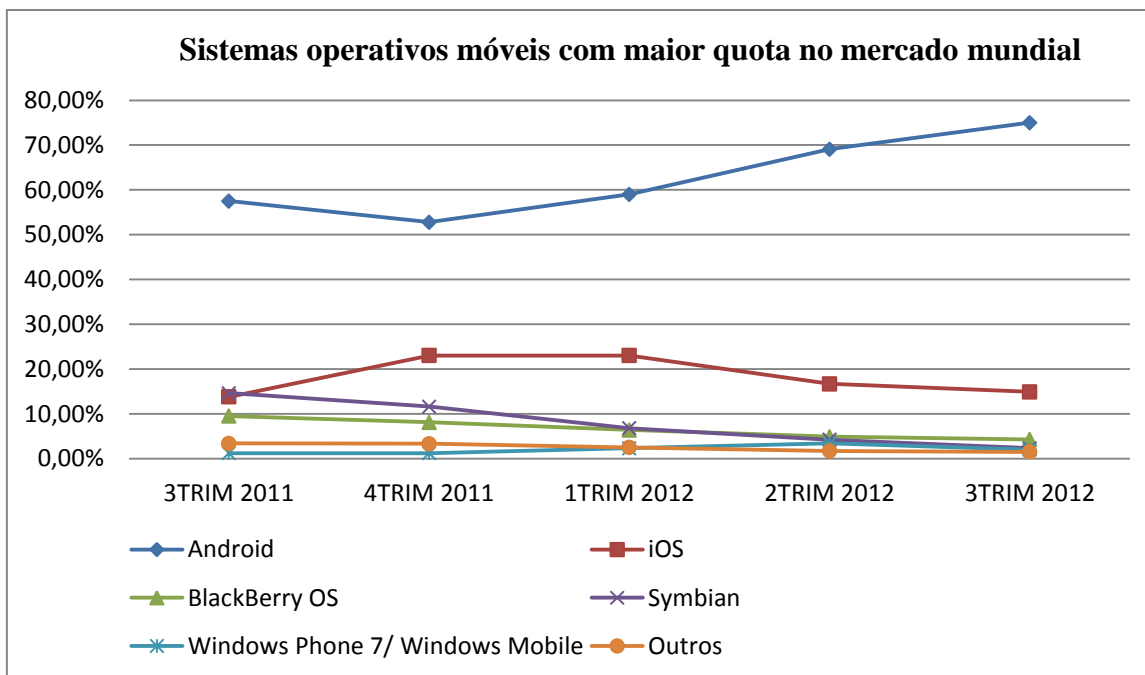
especificamente para correr em dispositivos móveis, como é o caso dos *smartphone* e *tablets*. Quando os programadores criam uma aplicação, estes escrevem e compilam a sua aplicação para um determinado sistema operativo. Isto deve-se ao facto que cada sistema operativo estar adaptado para comunicar com o *hardware* de forma própria e possuir uma API específica. Esta API baseia-se num conjunto de rotinas, protocolos e ferramentas para a construção de aplicações, facilitando o seu desenvolvimento ao fornecer blocos de construção ao programador (Darcey & Shane , 2012).

No mercado mundial existe uma enorme concorrência em relação à criação de aplicações ricas em recursos para dispositivos móveis, oferecendo uma melhor experiência e satisfação nas necessidades do usuário final. Atualmente existem cinco sistemas operativos principais no mercado europeu, são eles: Symbian (Accenture, 2013), iOS (Apple, 2013), Android (Google Inc./Opens Handset Alliance, 2013), BlackBerry OS (RIM, 2013) e Windows Phone (Microsoft, 2013).






O gráfico 2 apresenta os dados disponibilizados pela IDC no que diz respeito aos sistemas operativos móveis para *smartphones* mais vendidos no mercado a nível mundial, desde o terceiro trimestre de 2011 até ao terceiro trimestre de 2012. A tabela 3 apresenta o volume de *smartphones* com sistema operativo específico em milhões de unidades, no terceiro trimestre de 2011 e no respetivo trimestre de 2012. Os valores numéricos disponibilizados pela IDC, que deram origem ao gráfico que se segue, encontram-se discriminados na terceira tabela do Anexo A<sup>39</sup>.

---

<sup>39</sup> Dados percentuais, disponibilizados pela IDC, sobre quota do mercado mundial dos principais sistemas operativos móveis.



**Gráfico 2 - Quota do mercado mundial de sistemas operativos móveis para smartphones, entre o terceiro trimestre de 2011 e o terceiro trimestre de 2012.<sup>40</sup>**

Sistema Operativo	3TRIM 2011	3TRIM 2012	Comparação ano-a-ano
	(milhões de unidades)		
	71	136	91,5%
	17,1	26,9	57,3%
	11,8	7,7	-34,7%
	1,5	3,6	140%
	18,1	4,1	-77,3%
Outros	4,2	2,8	-33,3%
TOTAL	123,7	181,1	46,4%

**Tabela 4 - Volume de Smartphones vendidos com Sistema Operativo específico, a nível mundial, em milhões de unidades<sup>41</sup>**

<sup>40</sup> Fonte: Adaptado de RESTIVO *et al*, novembro 2012.

O sistema operativo móvel Android, da Google, dominou o terceiro trimestre do ano de 2012 no mercado mundial de *smartphones*, de acordo com o estudo efetuado pela IDC. Segundo com os dados conferidos pela empresa, o Android ocupou 75 por cento da quota de mercado, enquanto o iOS, da Apple, ficou pelos 14,9 por cento. Mais de 181 milhões de *smartphones* foram vendidos neste período, dos quais cerca de 136 milhões se encontravam equipados com Android. Três em cada quatro dos *smartphones* vendidos no mundo, durante o terceiro trimestre deste ano (de julho a setembro) são dispositivos Android. Na segunda posição encontra-se o sistema operativo móvel da Apple, iOS, com 14,9 por cento da quota de mercado e em terceiro o BlackBerry, da RIM, com 4,3 por cento. O iOS, sistema operativo do iPhone, desceu de 16,9 por cento no segundo trimestre, para os 14,9 do terceiro trimestre, com 26,9 milhões de *smartphones* vendidos. O Android, no período de um ano, apresentou um crescimento de 91,5 por cento, o que representa quase o dobro dos restantes sistemas operativos do setor.

### **3.4.1. Windows Phone**

Um dos sistemas operativos que já conta com muitos anos e que já se tornou uma antiguidade nos dias que correm é o Windows Mobile. Este SO foi muito utilizado nos dispositivos *Personal Digital Assistance* (PDA), uma versão PC do Windows e contemplava várias aplicações conhecidas, como o Word, Excel etc. Contudo, com o aparecimento de outros sistemas e dispositivos começou a ter o seu declínio. A Microsoft anunciou o Windows Phone 7 em 2010, sendo o sucessor do Windows Mobile. Surge então a parceria entre a Nokia e a Microsoft, com o intuito de se colocar entre os principais líderes do mercado (Costa, Duarte , & Filho, 2011)

A última versão é o Windows Phone 8, que estará disponível em dispositivos móveis em 2013, e que é compatível com o mais recente sistema operativo da Microsoft, o Windows 8, para computadores pessoais, portáteis, *notebooks* e *tablets*. A linguagem de programação usada nestes sistemas operativos é C#. O Windows Phone 8 está integrado com o *.NET FRAMEWORK* e conta com uma série de bibliotecas para desenvolvimento de aplicações (Microsoft , 2012).

---

<sup>41</sup> Fonte: Adaptado de RESTIVO *et al*, novembro 2012.

### 3.4.2. iOS

O iOS deriva da contração das palavras iPhone *Operating System*, surgiu em 2007 e é o sistema operativo móvel desenvolvido pela Apple (Ribeiro, 2012). É um sistema proprietário, pelo que não pode ser usado por *hardware* de terceiros. A Apple foi um dos primeiros grandes revolucionários na área dos *smartphones* com o seu iPhone, tendo a marca colocado este SO só nos seus dispositivos móveis (iPhone, iPad e iPod Touch) (Apple, 2012). O iOS, é uma versão modificada e otimizada do seu sistema operativo Mac OS X. A última versão deste sistema operativo é o iOS 6. Tem a grande vantagem de alto desempenho destes produtos, pela integração de *hardware* e *software* serem projetados um para o outro, nesta plataforma. Para fomentar o desenvolvimento de aplicações para este sistema operativo, é fornecido o iOS SDK, que contém o código, a informação e as ferramentas necessárias para desenvolver e otimizar aplicações (Ribeiro, 2012). A linguagem usada para desenvolver aplicações para o iOS é o Objective-C, que é uma linguagem desenhada para permitir uma programação orientada a objetos sofisticada. O Objective-C foi construído sobre a linguagem C, a maior parte da sua sintaxe deriva de C, enquanto os aspetos de orientação a objetos foram inspirados no estilo de passagem de mensagens do *Smalltalk*<sup>42</sup>. A linguagem foi escrita especificamente para trabalhar com o MacOS e adaptada para Linux e Windows, embora com limitações em algumas funcionalidades. A arquitetura do iOS é dividida em *Core OS* (núcleo do sistema operativo), *Core Services* (serviços oferecidos pelo sistema), *Media* (serviço de áudio, vídeo, fotos), *Cocoa Touch* (responsável pelas interações com o usuário) (Apple, 2012).

### 3.4.3. Android

O Sistema Operativo móvel Android é desenvolvido pela *Open Handset Alliance* (OHA). A OHA formou-se em novembro de 2007 com o intuito de criar um tipo de plataforma móvel com as melhores qualidades do mercado. É uma aliança de negócios composta por algumas das maiores e mais bem-sucedidas companhias de aparelhos móveis a nível mundial (Darcey & Shane, 2012). Entre os seus membros encontram-se fabricantes de *chpis*, telemóveis e programadores de *software*. Mais de metade dos seus membros são fabricantes de dispositivos: Samsung, Motorola, Dell, Sony Ericsson, HTC, LG, etc. e fabricantes de componentes elétricos e eletrónicos: Intel, NVIDIA, Texas Instruments, ARM e Qualcomm. Em outubro de 2008 o Android tornou-se uma plataforma *Open*

---

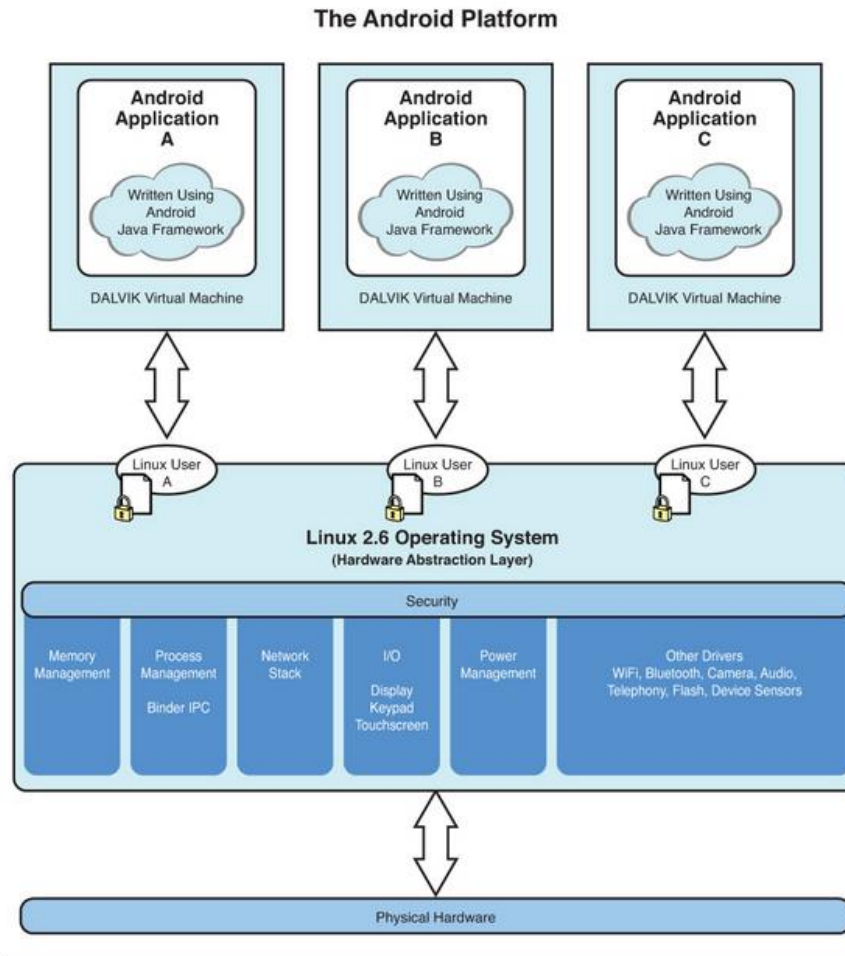
<sup>42</sup> *Smalltalk*, é uma linguagem de programação orientada a objeto dinamicamente.

*Source*<sup>43</sup>, sendo também apresentado o primeiro telemóvel com este sistema operativo (Ribeiro, 2012). Nem os programadores nem os fabricantes de dispositivos móveis necessitam de pagar taxas de licença para desenvolver aplicações para esta plataforma. Foi no terceiro trimestre de 2010 que o Android começou a dominar o mercado dos *SmatPhones* e tem vindo a ganhar terreno contra as já referidas plataformas móveis concorrentes. As aplicações para este sistema operativo são escritas em Java. Relativamente ao seu ambiente de desenvolvimento, possui um emulador, ferramentas de *debug*, memória e análise de performance. Para desenvolver aplicações pode ser usado o Eclipse SDK 4.2 (versão mais atual) através do *Plug-in Android Development Tools* (ADT). As aplicações Android podem ser desenvolvidas nos seguintes sistemas operativos: Windows XP (32 bits), Windows Vista (32-bit ou 64-bit), Windows 7 (32-bit ou 64-bit), Mac OS X 10.5.8 ou mais recentes (apenas x86), Linux (Ubuntu Linux 10.04 LTS Lucid Lynx, 8.04 LTS, ou posteriores) (Darcey & Shane , 2012) .

A plataforma Android foi projetada para ser mais tolerante a falhas do que muitos de seus antecessores. A licença do Android é flexível e permite que cada fabricante possa realizar alterações no código-fonte para normalizar os seus produtos, sem necessitar de compartilhar essas alterações com ninguém (Android, 2012). Relativamente a sua arquitetura, como se pode ver na figura 10, o dispositivo executa um sistema operativo Linux em que as aplicações são executadas de forma segura (Darcey & Conder, 2012).

---

<sup>43</sup> *Open Source* significa que é permitido desenvolver outros tipos de conceito semelhante com base nele



**Figura 10 - Arquitetura da plataforma Android<sup>44</sup>**

### **3.5. Equipamento de serviços de informação**

#### **3.5.1. Modelos de Base de Dados**

Uma Base de Dados é um sistema de armazenamento de dados relacionados entre si, de uma forma permanente, num sistema informático, com redundância controlada, acessíveis a um grupo de utilizadores e estruturado sob a forma de ficheiros de dados ou tabelas (Damas, 2005).

Para representarmos os dados ou estrutura de base de dados, também existem modelos específicos. Sendo que o mais conhecido e utilizado é o modelo relacional (Pereira & Poupa, 2011).

<sup>44</sup> Fonte: Retirado de DARCEY & CONDER (2012, p.30) .



O modelo relacional representa informações através de “relações” (ou tabelas). De acordo com o conceito do modelo, uma base de dados relacional é um conjunto de tabelas com número específico de atributos (colunas ou campos), e um número variável de tuplos<sup>45</sup> (Pereira & Poupa, 2011).

Damas (2005) classificou o modelo relacional em três gerações de sistemas, de acordo com os modelos utilizados, nomeadamente:

#### **1. Geração pré-relacional**

- Sistemas baseados em ficheiros;
- Modelo Hierárquico;
- Modelo em rede.

#### **2. Geração relacional**

- Modelo relacional.

#### **3. Geração pós-relacional**

- Modelo Orientado para objetos;
- Modelo Objeto- relacional;
- Outros modelos.

### **3.5.2. Características e Requisitos de um SGBD**

O sistema de gestão de base de dados (SGBD) é a única entidade que manipula a base de dados, são ainda a única entidade capaz de aceder diretamente à base de dados. Sendo que qualquer outro tipo de acesso terá de ser realizado pedindo o serviço ao SGBD (Miranda, 2010). O SGBD fornece um nível intermédio entre o utilizador e os dados. Ou seja, num SGBD, o utilizador (pessoa ou aplicação) solicita um serviço que pretende e é o próprio SGBD que faz todo o conjunto de acessos de modo a responder a esse pedido (Damas, 2005). Assim, segundo Damas (2005), deve-se optar por SGBD quando:

- Informação for armazenada de modo permanente;
- Controlo central de dados;
- Controlo de redundância;
- Consistência e integridade dos dados;
- Múltiplos utilizadores;
- Independência dos dados das aplicações;

---

<sup>45</sup> Tuplos: representação de cada linha de uma base de dados, formada por um conjunto de colunas.

- Partilha de dados;
- *Backup & Recovery*.

O objetivo de um SGBD é fornecer um ambiente apropriado para aceder e armazenar informação de dados de forma fiável e eficiente, deste modo um SGBD tem de cumprir importantes requisitos para atingir tal objetivo. Requisitos que, de acordo com Damas (2005), são:

- Eficiência – ser capaz de aceder, processar e alterar grandes volumes de dados de forma eficiente;
- Robustez – Manter os dados de forma consistente, mesmo após falhas de *hardware* ou erros de *software*;
- Controlo de Acessos – Controlar o acesso de múltiplos utilizadores aos dados de forma consistente e apenas a utilizadores autorizados;
- Persistência – Manter dados durante longos períodos, independentemente das aplicações que a eles fazem acesso.

Deste modo, o SGBD a implementar na arquitetura alvo terá de cumprir os requisitos supramencionados.

Existem diversas escolhas no mercado de SGBD e dividem-se em dois grandes grupos, grande dimensão e pequena dimensão, são exemplos:

- Grande dimensão: ORACLE, Microsoft SQL Server, Ingres, Informix e DB2;
- Pequena dimensão: MySQL, Dbase, FoxPro e Microsoft Access.

Segundo Pereira & Poupá (2011) os SGBD mais utilizados no mercado, no que concerne ao desenvolvimento de aplicações web/móvel, são: Oracle, Access, PostgreSQL, MySQL e SQL Server. Serão abordados nas secções que se seguem, três dos sistemas mencionados.

### **3.5.3. Access**

O Access, também conhecido por MS Access, é um SGBD da Microsoft, incluído no pacote do Microsoft Office. Permite o desenvolvimento rápido de aplicações que envolvem tanto os modelos e estrutura de dados, como também a interface a ser utilizada pelos utilizadores (Microsoft, 2012).

O Access é capaz de usar dados guardados em Access/Jet, SQL Server, Oracle, ou qualquer sistema de base de dados compatível com ODBC<sup>46</sup>.

Com o Access é possível desenvolver desde aplicações simples, tais como um cadastro de clientes, controlo de pedidos, ou aplicações mais complexas, como por exemplo, todo o controlo operacional, administrativo e financeiro de uma pequena ou média empresa (Microsoft, 2012).

### 3.5.4. SQL Server

O MS SQL Server é um SGBD relacional criado pela Microsoft em parceria com a Sybase<sup>47</sup> em 1988 e inserido como produto complementar do Windows NT. No final da parceria, em 1994, a Microsoft continuou aperfeiçoando o produto. O MS SQL Server funciona apenas nas várias versões do sistema operativo Windows, da Microsoft, ao contrário dos seus grandes concorrentes, Oracle e Postgres, que funcionam em diversas plataformas e sistemas operativos diferentes. Utiliza o *standard ANSI*<sup>48</sup> SQL e um “dialecto” proprietário designado por *Transact SQL*<sup>49</sup> (Almeida, 2011).

### 3.5.5. MySQL

O MySQL<sup>50</sup> é um sistema de gestão de base de dados relacionais, ou seja, um sistema informático com o objetivo de armazenar dados de forma bem organizada e com baixa redundância. Tal como os outros sistemas de base de dados atuais, funciona como um sistema cliente-servidor. Onde o programa servidor é responsável por armazenar e permitir a manipulação de dados. Os programas cliente possibilitam o acesso à base de

---

<sup>46</sup> ODBC, *Open Data Base Connectivity* é uma norma para acesso ao SGBD. Esta norma define um conjunto de interfaces que permitem o uso de linguagens de programação capazes de utilizar estas interfaces, para ter acesso a uma vasta gama de bases de dados distintas sem a necessidade de codificar métodos de acesso especializados.

<sup>47</sup> Sybase, uma empresa SAP, é uma empresa de *software* que produz serviços e produtos relacionados à gestão de informação, mobilidade, *messaging*, ferramentas de desenvolvimento e *data warehousing* e OLAP.

<sup>48</sup> Acrónimo para *American National Standards Institute*

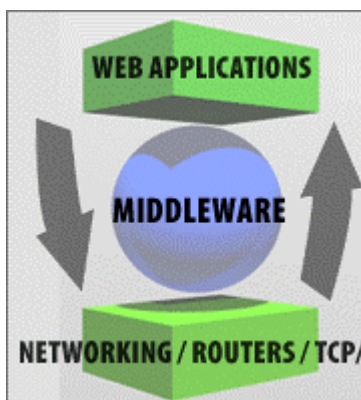
<sup>49</sup> *Transact-SQL* (T-SQL) é propriedade da Microsoft e Sybase para a extensão SQL, aperfeiçoa a linguagem SQL no que concerne a: Variáveis Locais; Várias funções de suporte ao processamento de strings, datas, funções matemáticas, etc ; as instruções DELETE e UPDATE.

<sup>50</sup> O MySQL é de utilização gratuita e pode ser obtido em: <http://dev.mysql.com>

dados do ponto de vista lógico, e comunicam com o servidor utilizando a linguagem SQL<sup>51</sup>, concretamente o ANSI/ISO SQL *standard*, para manipulação dos dados e das estruturas onde são armazenados (Pereira & Poupa, 2011).

### 3.6. Serviços de Comunicações

Os serviços de comunicações são serviços fixos, privados e de interesse coletivo, permitindo a oferta de informação multimédia através de qualquer meio de comunicação. Podem ser considerados essencialmente por um serviço de rede. Cada serviço de rede é composto, no mínimo, por quatro elementos: servidor, cliente, protocolo e *middleware*<sup>52</sup> (Kurose & Ross, 2007). O servidor é o computador que realiza a parte principal do serviço, através de recursos locais e/ou outros serviços. O cliente é o computador que solicita o serviço através da rede (Kurose & Ross, 2007). Normalmente, o cliente age a pedido de um ser humano, através de uma interface de utilizador, embora também possa ser representado por outro sistema computacional. O protocolo é a definição do serviço propriamente dito, ou seja, os passos, o conjunto de mensagens e os formatos de dados que definem o diálogo necessário entre o cliente e o servidor para a realização do serviço. O *middleware* é o suporte de execução e de comunicação que permite a construção do serviço. Este é composto por sistemas operativos e protocolos de rede que são os responsáveis por enviar os pedidos do cliente para o servidor e a resposta de volta ao cliente, conforme é ilustrado na figura 11 (Pinheiro, 2010).



<sup>51</sup> SQL, acrónimo para *Structed Query Language*, trata-se de uma linguagem originalmente desenvolvida pela IBM a partir do modelo relacional proposto por Edgar Frank Codd, em 1970.

<sup>52</sup> *Middleware* que significa mediador, designa camadas de *software* que não constituem aplicações, mas facilitam o uso de ambientes ricos em tecnologias de informação;

### **3.6.1. Rede Ethernet**

A tecnologia *Ethernet*, nas suas múltiplas variantes, aparece como uma tecnologia dominante no mercado, no que concerne a rede de utilizadores ou de distribuição. Esta tecnologia foi desenvolvida em diversas fases pela Xerox, Intel e DEC, nos meados da década de 1970. O seu desenvolvimento foi iniciado pela Xerox, por Robert Metcalfe e David Boggs, posteriormente foi normalizado pelo *Institute of Electrical and Electronic Engineer* (IEEE), como norma IEEE 802.3. Inicialmente desenvolvida para redes locais que, hoje em dia, têm sido utilizada noutras áreas e têm uma aceitação no mercado quase universal. Considera-se uma das tecnologias de rede mais utilizadas em todo mundo, sendo que uma grande parte do tráfego da Internet tem origem ou destino em redes *Ethernet* (Pinheiro J., 2010).

#### **Características de uma rede Ethernet**

Ethernet é uma tecnologia de barramento de difusão de 10 Mbps, 100 Mbps e até 1Gbps, com método de entrega sem garantia e controlo de acesso distribuído. É um barramento porque todas as estações partilham um único canal de comunicação; é de difusão (*broadcast*) porque todos os transdutores<sup>54</sup> recebem cada uma das transmissões (Boavida & Bernardes, 2012).

O controlo de acesso à *Ethernet* é distribuído porque, ao contrário de algumas tecnologias de rede, a *Ethernet* não possui nenhuma autoridade central para permitir o acesso. O método de acesso à *Ethernet* é chamado *Carrier Sense Multiple Access with Collision Detect* (CSMA/CD). É um CSMA porque vários equipamentos podem aceder a *Ethernet* simultaneamente e cada um deles estabelece se o meio está ou não, detetando a presença ou não de sinal. Quando uma interface do *host* tem um pacote para transmitir, ela verifica o meio para saber se há alguma mensagem a ser transmitida (ou seja, deteta o transceptor). Se nenhuma transmissão for detetada, a interface do *host* inicia a transmissão (Boavida & Bernardes, 2012, p. 85) . Cada uma das transmissões possui um limite de

---

<sup>53</sup> Retirado de <http://www.networkworld.com/community/node/32389>.

<sup>54</sup> Transceptor (abreviação de transmissor – recetor) é um dispositivo que permite a transmissão de dados entre diferentes sistemas de cablagem.

duração, porque há um tamanho máximo de pacote de dados com 1500 *bytes*, designado por *maximum transmission unit* (MTU). E caso o nível acima tenha um datagrama superior a 1500 *bytes* tem que dividir antes de o passar para o nível de ligação de dados. Além do mais, o *hardware* deve observar um intervalo mínimo de tempo entre as transmissões, o que significa que nenhum par de equipamentos, que comuniquem entre si, pode utilizar a rede sem oferecer aos demais equipamentos uma oportunidade de acesso (Boavida & Bernardes, 2012, p. 89).

Quando existe colisão, a *Ethernet* lida com este evento de uma forma simples. Cada transceptor faz a monitorização do cabo. Enquanto transmite informação, verifica se algum sinal interfere na transmissão. Esta verificação recebe o nome de CD (*collision detect*), tornando a *Ethernet* numa rede de CSMA/CD. Quando uma colisão é detetada, a interface do *host* interrompe a transmissão, espera que a atividade cesse e tenta reinicia-la (Santos , 2010).

A *Ethernet* utiliza uma política chamada *binary exponencial backoff*, na qual um transmissor espera um tempo aleatório após a primeira colisão (Santos, 2010). A duração máxima deste tempo aleatório é duas vezes a da primeira tentativa, caso a segunda tentativa também resulte em colisões; quatro vezes caso a terceira tentativa também resulte em colisão, e assim por diante (Santos , 2010).

O motivo para o uso do conceito *binary exponencial backoff* é que, numa situação improvável na qual muitas estações tentam transmitir simultaneamente, pode ocorrer um grande congestionamento. Em tal situação, há grande probabilidade de duas estações escolherem intervalos aleatórios que estejam próximos uns dos outros. Por isso, a probabilidade de uma outra colisão ocorrer é muito grande. Aumentando para o dobro o tamanho máximo do retardo aleatório de transmissão, a estratégia *binary exponencial backoff* dispersa rapidamente as tentativas de retransmissão das estações num período de tempo razoavelmente longo, diminuindo, assim, a probabilidade de ocorrência de futuras colisões (Santos, 2010, p. 97).

### **A capacidade da *Ethernet***

A necessidade crescente de capacidade de transmissão levou ao desenvolvimento da tecnologia *Ethernet* a 100Mbps, normalmente denominada *Fast Ethernet*, especificada na norma 802.3u. Além de possibilitar um débito consideravelmente superior ao débito da *Ethernet* padrão, esta tecnologia apresenta outras vantagens, das quais se destacam: o baixo custo, a capacidade de auto negociação do débito a utilizar (10 ou 100Mbps) e o

funcionamento em *full duplex* e em modo comutado que, em conjunto resolvem o problema das colisões. Entre 1996 e 1999, decorreu o trabalho de desenvolvimento de especificações para *Ethernet* de 1Gps, no âmbito da *task force* 802.3z do IEEE e da, então designada, *Gigabit Ethernet Alliance* (GEA), da qual fizeram parte as companhias como a 3Com, Bay Networks, Cisco, Compaq, Intel e Sun (Boavida & Bernardes, 2012, p. 91).

A tabela que se segue mostra resumidamente algumas das variantes de *Ethernet* a 100Mbps e 1Gps.

Características	100-Base-TX	1000-Base-SX	1000-Base-LX	1000-Base-T
<b>Velocidade</b>	100 Mbps	1 Gbps	1 Gbps	1 Gbps
<b>Comprimento máximo dos seguimentos</b>	100m	220m; 550m	550m; 5Km	<100m
<b>Meio Físico</b>	UTP cat5 dois pares	Fibra ótica multimodo	Fibra ótica multimodo ou monomodo	UTP, Cat5, quatro pares
<b>Utilização</b>	Generalizada	Em <i>backbones</i> curtos	Em <i>backbones</i> de edifício	Interligação de <i>clusters</i> de equipamentos e postos de trabalho.

Tabela 5 - Características de algumas variantes de Ethernet<sup>55</sup>

### 3.6.2. Redes Sem Fios

Uma rede sem fios, também chamada rede *wireless*<sup>56</sup>, é um sistema que interliga vários equipamentos fixos ou móveis, e utiliza o ar como meio de transmissão. Semelhante a uma rede local que usa fios convencionais (cabo coaxial, par trançado e fibra ótica), a rede sem fios usa ondas de rádio. Os dados a serem transmitidos são modulados numa portadora e transmitidos através de ondas eletromagnéticas. Podem existir múltiplas

<sup>55</sup> Fonte: Boavida & Bernardes, 2012

<sup>56</sup> *Wireless* (*wire*: fio/cabo; *less*: sem), significa rede sem fios. Refere-se a uma rede de dispositivos sem a necessidade de uso de cabos, as comunicações são feitas via ondas de rádio ou via infravermelhos.

portadoras num mesmo meio sem que uma interfira com outra. O recetor, para extrair os dados, sintoniza a frequência específica da portadora e rejeita as outras frequências diferentes (Silva, 2012).

### **Tipos de Redes Sem Fios**

Segundo Silva (2012), as redes sem fios podem ser de vários tipos:

- Redes pessoais (WPAN - *Wireless Personal Area Network*);
- Redes locais (WLAN - *Wireless Local Area Network*);
- Redes metropolitanas (WMAN - *Wireless Metropolitan Area Network*);
- Redes de áreas alargadas (WWAN - *Wireless Wide Area Network*)

As redes pessoais sem fios, chamadas de redes WPAN, são usadas para interligar dispositivos eletrónicos fisicamente próximos. Este tipo de rede pode interligar teclados, impressoras, máquinas fotográficas, ratos, etc. Nos equipamentos mais recentes é utilizado o protocolo *Bluetooth*<sup>57</sup> para estabelecer esta comunicação (Boavida & Bernardes, 2012).

As redes sem fio do tipo WLAN referem-se à comunicação de equipamentos em áreas restritas. É uma rede local que usa ondas de rádio para fazer uma ligação à rede de suporte. Inicialmente as redes WLAN eram muito caras e foram somente usadas como uma alternativa ao LAN-Internet com cabo, nos lugares onde instalar cabos era difícil ou impossível (Kushki, Plataniotis, & Venetsanopoulos, 2012).

A “popularidade” dessa tecnologia deve-se ao protocolo IEEE802.11 (IEEE, 2012). Este protocolo, também é chamado de Wi-Fi, marca registrada por um grupo de fabricantes que se uniram para criar um selo de garantia de compatibilidade (Wi-Fi Alliance, 2012).

As redes metropolitanas sem fios, WMAN, referem-se a redes que ocupam o perímetro de uma cidade. O IEEE 802.16, ou *Worldwide Interoperability for Microwave Access* (WiMAX) tem como objetivo disponibilizar o acesso banda larga sem fios para novas localizações de modo a cobrir maiores distâncias (Wimax, 2012).

As redes sem fio de áreas alargadas, ou WWAN, abrangem uma grande área geográfica, como um país ou continente. Estes tipos de redes tornaram-se necessários devido ao crescimento das empresas que precisam de passar informação entre empresas de forma rápida e eficiente (Lima, Soares, & Endler, 2004).

A figura 12 destaca a família de padrões IEEE 802 para as redes sem fios.

---

<sup>57</sup> *Bluetooth* é o nome de uma tecnologia de comunicação sem fios que interliga e permite a transmissão de dados entre dispositivos eletrónicos através de ondas de rádio.



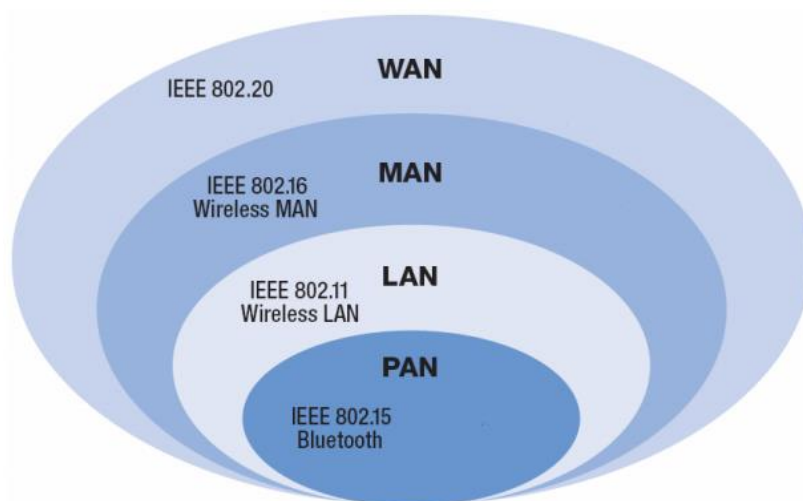


Figura 12 - Padronização global IEEE para redes sem fio<sup>58</sup>

### Topologia da Rede Wi-Fi

As WLANs estabelecem comunicação de dados entre os pontos da rede. O protocolo IEEE 802.11 controla a maneira como os dados são transportados através da camada física, e define uma subcamada *Media Access Control*<sup>59</sup> (MAC), que manipula a interface entre a camada física e a restante estrutura da rede. A norma de rede sem fios IEEE 802.11 é identificada por um *Service Set Identifier* (SSID), atribuído pelo administrador da rede. É através do mesmo que, quando configurado como visível, os utilizadores têm acesso a lista de redes sem fios disponíveis num determinado local. Existem várias versões da norma IEEE 802.11, sendo as seguintes as mais utilizadas no mercado: IEEE 802.11a, IEEE 802.11b, IEEE 802.11g e IEEE 802.11n (Boavida & Bernardes, 2012).

Boavida & Bernardes (2012) fazem a descrição das seguintes variantes da norma IEEE 802.11:

#### • 802.11a

Esta versão da norma IEEE 802.11 funciona na frequência de 5GHz. No entanto utilização desta norma está condicionada por dois fatores a ter em conta: a frequência utilizada e a incompatibilidade com as outras normas IEEE 802.11.

<sup>58</sup> Adaptado de LIMA *et al.* (2004, p.8)

<sup>59</sup> *Media Access Control*, significa controlo de acesso ao meio. Termo utilizado em redes de computadores para designar a subcamada da camada de ligação de dados, camada 2 do modelo OSI (*Open Systems Interconnection*).

### • 802.11b

Esta norma funciona na frequência 2.4GHz e tem um ritmo de transferência de dados que pode alcançar os 11Mbps. A *Wireless Ethernet Compatibility Alliance* (WECA) deu a esta norma a referência *Wi-Fi*, assim qualquer dispositivo que tenha esta denominação dá a garantia que é compatível com outros componentes da norma 802.11.

### • 802.11g

Pode-se afirmar que esta norma é uma extensão da norma 802.11b, ou seja, são semelhantes, no entanto a taxa de transferência de dados desta norma vai até os 54Mbps. Esta norma tem a seu favor o facto de ser compatível com a norma 802.11b, sendo possível ter vários componentes interligados com as duas normas. Porém, neste caso a rede vai comunicar a velocidade mais baixa (11Mbps).

### • 802.11n

Esta norma foi aprovada pelo IEEE recentemente, e tem a vantagem de funcionar nas duas frequências (2.4GHz e 5GHz), desta forma podem alcançar velocidades entre os 65Mbps e os 600Mbps, desde modo é a norma das redes sem fio mais utilizada para distribuição multimédia.

Para haver compatibilidade com as outras normas, esta norma utiliza o sistema *Multiple Input, Multiple Output* (MIMO). O MIMO permite que uma placa utilize fluxos de transmissão através de vários conjuntos transmissores, recetores e antenas permitindo assim efetuar a transmissão de forma paralela.

Assim, é possível resumir as características das normas 802.11 na seguinte tabela:

Versões	Banda (GHz)	Número de fluxos MIMO	Debito Máximo (Mbps)
802.11	2.4	1	1-2
802.11b	2.4	1	11
802.11a	5	1	54
802.11g	2.4	1	54
802.11n	2.4/5	4	600

Tabela 6 - Variantes da norma IEEE 802.11<sup>60</sup>

---

<sup>60</sup> Retirado de Boavida & Bernardes (2012, p. 99)

Segundo Silva (2012), as redes WLAN podem ser configuradas de dois modos diferentes: Redes Infraestruturadas e Redes *Ad-Hoc*<sup>61</sup>. As Redes Infraestruturadas têm como característica possuir pelo menos dois tipos de elementos: as estações e os pontos de acesso. Os pontos de acesso são responsáveis pela interligação das estações móveis com a rede fixa. Cada ponto de acesso tem o controlo de uma determinada área de cobertura e realiza tarefas importantes de coordenação das estações móveis na sua área (ver figura 13).

As redes *Ad-Hoc* têm como característica não possuir nenhuma infraestrutura associada para apoiar a comunicação. Os diversos equipamentos móveis ficam localizados numa pequena área onde estabelecem comunicação ponto-a-ponto por um certo período de tempo (Silva, 2012).

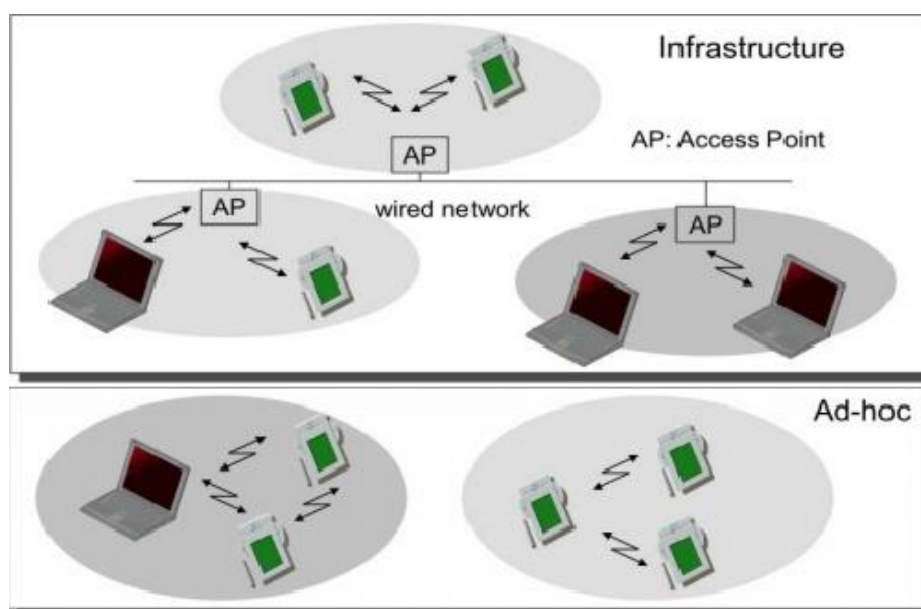


Figura 13 - Modos de configuração básicos na tecnologia Wi-Fi<sup>62</sup>

### Vantagens e Desvantagens

A principal vantagem das redes sem fios é permitir ao utilizador mobilidade, mantendo-o ligado à rede em qualquer lugar. Esta característica elimina ainda a necessidade de passar cabos por tetos, paredes, etc. Proporcionando, assim, que a rede possa alcançar locais de difícil acesso. Têm menor necessidade de manutenção, maior

<sup>61</sup> Ad-Hoc é uma expressão latina que significa "para esta finalidade" ou "com este objetivo". Refere-se a uma solução de carácter temporário, destinada resolver um problema imediato.

<sup>62</sup> Retirado de Osório (2011, p. 31)

robustez e permitem a vários equipamentos trabalhar ao mesmo tempo utilizando a mesma gama de frequências (Korowajczuk, 2011).

Esse tipo de redes apresenta também alguns inconvenientes, como por exemplo o consumo de energia dos dispositivos móveis, ou o facto do meio físico de transmissão ser de domínio público, propício a interferências e problemas de segurança. As características do meio podem ainda influenciar a propagação do sinal, tais como a introdução de fenómenos de interferência e de *fading*<sup>63</sup>, que podem conduzir a variações no raio de alcance e no débito máximo disponível, restringindo a conectividade e o desempenho da rede (Korowajczuk, 2011).

### 3.7. Conclusões

Foram abordados três servidores, o Apache, IIS e o Apache Tomcat. A escolha do servidor ideal para ser implementado na dissertação depende dos objetivos da aplicação. Como foi decidido adotar *software* de código aberto, o IIS sai a perder em relação aos restantes servidores abordados. Já entre o Apache e o Apache Tomcat, ambos de código aberto, a escolha foi feita tendo em conta qual deles é o mais adequado para a página Web que se pretende desenvolver. Como a ideia é desenvolver páginas dinâmicas a escolha recai para o Apache Tomcat, pois segundo DevMedia (2011), o Tomcat responde a pedidos de páginas dinâmicas.

No âmbito de linguagens para aplicações na Web, descreveu-se a linguagem XML e analisou-se a linguagem HTML. Foram igualmente analisadas as características que sustentam a evolução da linguagem HTML que deu origem à versão HTML4, incluindo folhas de estilo CSS que permitem separar o conteúdo do aspeto.

Contudo, as aplicações multimédia na Web necessitam de funcionalidades mais complexas que permitam ampliar a gama de elementos interativos na Web e, por outro, gerar uma página HTML, dinamicamente, no lado do servidor, que inclua conteúdos adaptados às necessidades do utilizador e que podem ser obtidos, neste caso, a partir de uma base de dados. Para aumentar a interatividade das páginas HTML, pode-se recorrer ao desenvolvimento de scripts, ou de objetos, que são inseridos nas páginas HTML e executados no lado do cliente, ou seja, pelo *browser*. Neste contexto, analisou-se as novas funcionalidades que foram introduzidas na linguagem HTML5.




---

<sup>63</sup> *Fading* refere-se ao desvio da atenuação que um sinal de telecomunicação de frequência modulada pelo portador experimenta sob certos meios de propagação.

Também foram abordados as tecnologias que permitem desenvolver páginas Web, dando destaque a três mais utilizadas no mercado JSP, ASP e PHP. As três apresentam inúmeras vantagens e alguns inconvenientes, que também foram abordados.

O ASP.NET, sendo propriedade da Microsoft, faz com que a criação de uma página Web utilizando esta tecnologia, obrigue o programador a ter de utilizar os produtos e linguagens da empresa proprietária. O PHP tem a vantagem de ser compatível com vários sistemas de gestão de base de dados e sistemas operativos. No entanto, a sua linguagem ainda é desconhecida no mundo académico. O JSP, além de ser de código aberto, utiliza uma linguagem de programação muito conhecida pelos estudantes e a sua linguagem de *script* é das mais utilizadas na internet. No entanto, tem a desvantagem de ser compatível apenas com um sistema de base de dados, e ter de recorrer a *drivers* para poder ter acesso a outros sistemas de base de dados. Como o objetivo, no que concerne a aplicação web, é desenvolver uma página web dinâmica, a tecnologia JSP cumpre com este requisito. Pois permite implementar o HTML, XML e ainda o CSS, além de ter a vantagem de ser de código aberto, que é um dos critérios preponderantes para a escolha das tecnologias e programas a implementar neste projeto.

Foram ainda analisados os três dos principais sistemas operativos utilizados em dispositivos móveis. Como súmula, apresenta-se de seguida uma tabela com as principais características destes sistemas operativos móveis: Windows Phone, iOS e Android.

Características			
<b>Última versão</b>	Windows Phone 8	iOS 6	Android 4.1
<b>Empresa</b>	Microsoft	Apple	Google
<b>Núcleo</b>	Windows NT	OS X	Linux
<b>Plataforma</b>	Windows	MAC OS	Windows
<b>SDK</b>			MAC OS
			Linux
<b>Hardware</b>	Pouca variedade	Apenas iPhone e iPad	Grande variedade
<b>Vantagens</b>	Bom desempenho	Melhor sensibilidade <i>touchscreen</i>	<i>Open Source</i>
	Múltiplos fabricantes	<i>Design</i> atrativo	Múltiplos fabricantes
			Custo acessível
<b>Inconvenientes</b>	Suscetível a <i>malware</i> <sup>64</sup>	Custo elevado	Suscetível a <i>malware</i>

**Tabela 7 - Comparação de Sistemas Operativos Móveis**

Através desta tabela comparativa verifica-se que cada plataforma tem as suas mais-valias e os seus pontos fracos. É difícil dizer com certeza qual a melhor plataforma para desenvolver uma aplicação. Cada programador geralmente aconselha aquela em que esta atualmente a desenvolver aplicações. No entanto, tal como verifica-se nos dados da IDC, o sistema operativo com maior quota no mercado mundial é o Android, que aliado ao facto de ser *open source*, possui uma grande variedade de *hardware* a preços acessíveis, será o sistema eleito para o desenvolvimento deste projeto.

Hoje em dia, a organização da informação é um dos aspetos preponderantes para a partilha da mesma numa organização, e uma das formas mais utilizada para organizar a mesma são as bases de dados. Tendo em conta que a tecnologia informática evoluiu de tal

<sup>64</sup> *Malware* (*malicious software*) é um *software* destinado a se infiltrar num sistema alheio de forma ilícita, são exemplos os vírus de computador, *worms*, *trojan horses* (cavalos de troia) e *spywares*,

forma que é possível organizar as bases de dados com tecnologias denominadas sistemas gestores de base de dados. Destes sistemas, foram abordados os mais utilizados no mercado (Acess, SQL Server e MySQL). A implementação dos mesmos depende muito da finalidade do SGBD e das tecnologias associadas que serão utilizadas. Dos três SGBD supramencionados destaca-se o MySQL, pelo facto de ser de código aberto e compatível com diversos sistemas operativos (Windows, Linux, Solaris, Sun OS). Deste modo, é o SGBD adequado para os propósitos da presente dissertação.

As redes sem fios oferecem conectividade em áreas onde a conexão por cabos é difícil ou até mesmo impossível. Ao mesmo tempo possibilita que equipamentos e aplicações móveis possam operar com os critérios de mobilidade e flexibilidade exigidos para este trabalho. As redes Wi-Fi são muitas vezes implementadas para servirem de ponte entre uma rede convencional e um grupo de utilizadores remotos. São bastante simples de configurar para interligar computadores e dispositivos móveis, desde que sejam compatíveis e estejam geograficamente próximos. É um sistema prático e eficaz, que elimina cablagem e permite facilmente a adesão de novos utilizadores, pelo que, associado a uma rede Ethernet convencional, será a melhor opção a aplicar na camada de sistemas de comunicação da arquitetura deste projeto.



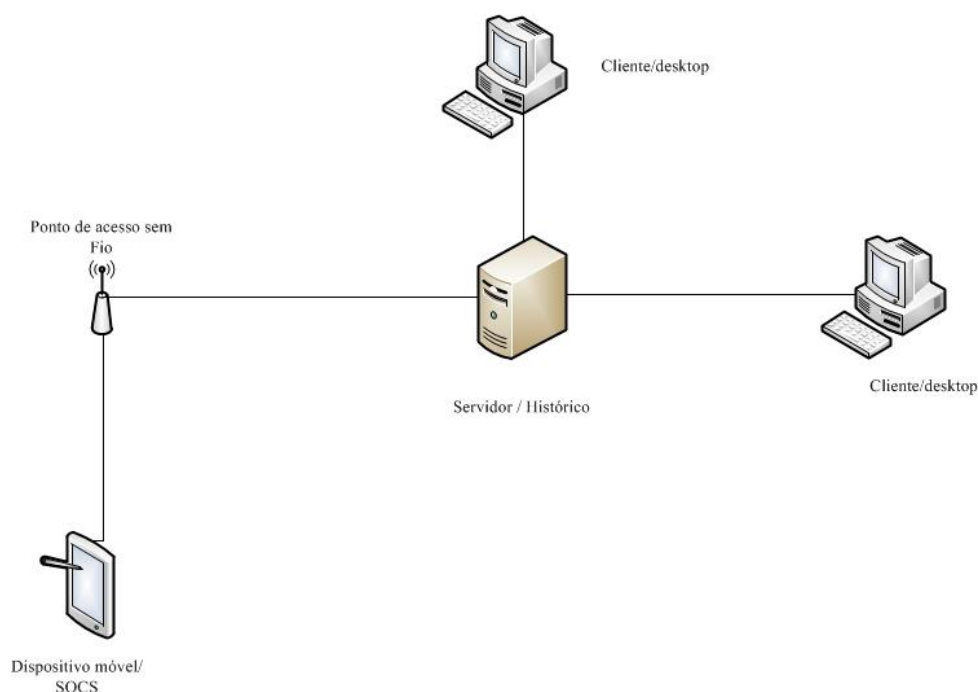


## CAPÍTULO IV DESCRIÇÃO DAS APLICAÇÕES

### 4.1. Introdução

Tendo em conta os objetivos traçados na presente dissertação, relativamente ao sistema proposto, é importante definir os requisitos das aplicações, que servirá como guia para a conceção das mesmas. Assim, é importante descrever o que se quer com cada uma das aplicações, e se é possível atingir. Desta forma, os requisitos que serão especificados estabelecem o modo como a aplicação funcionará, bem como o aspeto que a mesma terá.

Na figura 14 é possível identificar cada um dos elementos que definem o sistema, e será o ponto de partida para a definição dos requisitos das aplicações. Para cada um dos elementos especificados na figura, exceto o ponto de acesso sem fio, são definidos os requisitos do utilizador, os quais compreendem as funcionalidades oferecidas pelo sistema e os requisitos do sistema que detalham as mesmas.



**Figura 14 - Diagrama do sistema a implementar**

O sistema proposto funcionará dentro do navio, e é constituído pelos dispositivos apresentados na figura anterior, nomeadamente: os clientes fixos, que são constituídos pelos computadores *desktop* com acesso ao servidor de páginas *web*; os clientes móveis, que agrupa todos os dispositivos móveis que contenham a aplicação a ser desenvolvida; o

servidor, é o computador onde se encontram instalados os *web services* com acesso a base de dados (com histórico das ações) e o servidor web.

No que concerne aos utilizadores, o sistema pode ser dividido em três módulos, nomeadamente:

- Aplicação móvel, o utilizador poderá efetuar as atividades fornecida pela mesma, e enviar informação para base de dados;
- Aplicação Web, nesta aplicação será feita a consulta dos resultados das atividades realizadas pela aplicação móvel, que se encontram guardados na base de dados;
- Base de Dados, sistema que vai armazenar os dados de cada atividade realizada, enviada através da aplicação móvel. Para posterior consulta pela aplicação web.

De acordo com a escolha da tecnologia a implementar para a solução, optou-se por *software* livre. Como foi definido no capítulo dois e capítulo três, a aplicação móvel será desenvolvida para plataformas Android. A mesma aplicação deverá comunicar com o SGBD MySQL, através do servidor web Apache Tomcat, com o objetivo de obter o produto final do referido projeto. No que concerne a aplicação web, a mesma será feita utilizando a tecnologia JSP.

## **4.2. Requisitos da Aplicação móvel**

Após a descrição do sistema e dos módulos que constituem o mesmo, nesta secção será analisada os requisitos da aplicação móvel. Neste caso, como já definido na presente dissertação, trata-se duma aplicação que vai permitir observar e controlar o estado de execução das SOC's do navio<sup>65</sup> através dum dispositivo móvel. Deste modo, é importante perceber como estão organizadas/distribuídas as SOC's no navio.

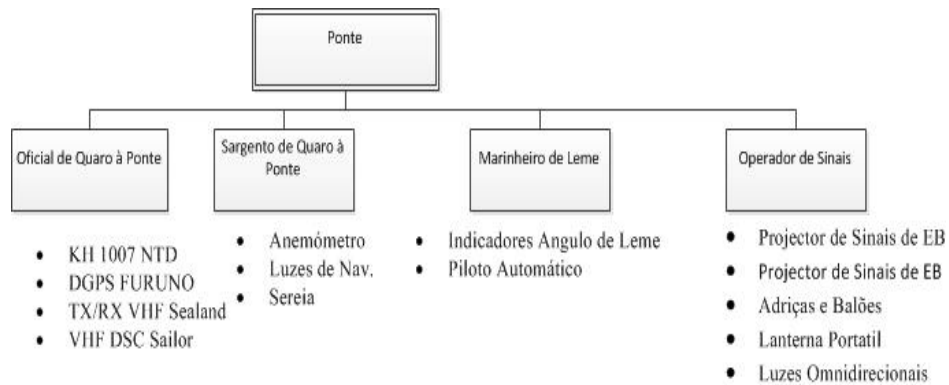
De acordo com IONAV 60, as SOC's nos navios estão organizadas de acordo com as funções dos elementos da guarnição (Bué, 2012). Logo, como foi definido na análise do problema, o protótipo funcional a implementar será definido para os elementos que desempenhem funções na ponte. Desta forma, para aplicação móvel, serão analisadas as SOC's atribuídas aos seguintes elementos da ponte: Oficial de quarto à ponte, Sargento de Quarto à ponte, Marinheiro do leme e o Operador de sinais.

Tendo sido determinado quais os elementos de quarto à ponte que são potenciais utilizadores deste sistema, é necessário identificar quais os equipamentos atribuídos a cada elemento de quarto à ponte para que seja possível numa fase posterior testar a bordo

---

<sup>65</sup> Para esta dissertação, os navios alvos são as Fragatas da Classe Vasco da Gama da Marinha Portuguesa.

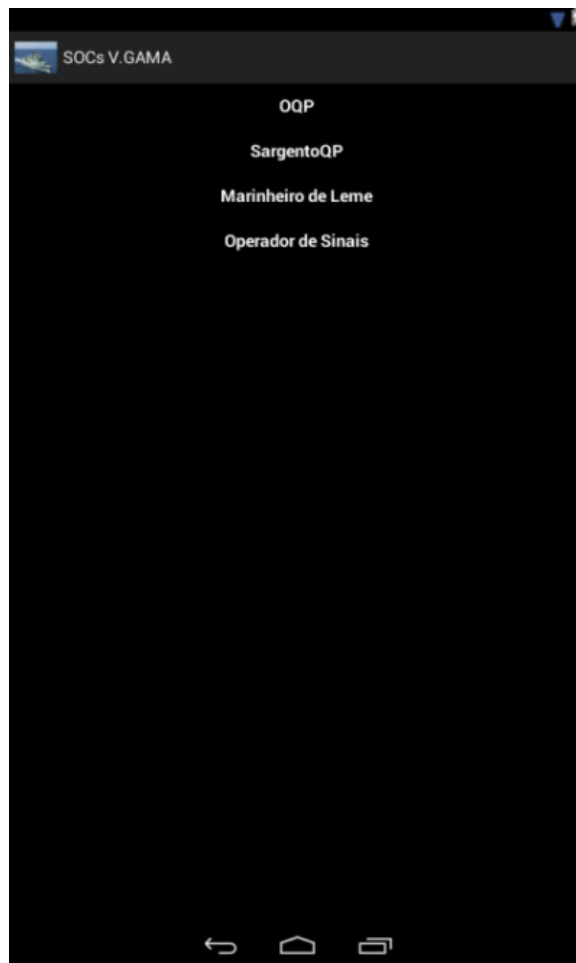
exequibilidade da aplicação que se pretende implementar. Na figura 15 é apresentado um diagrama que afeta a cada elemento de quarto à ponte os equipamentos que tem que verificar.



**Figura 15 - Diagrama dos Equipamentos a Implementar na Aplicação**

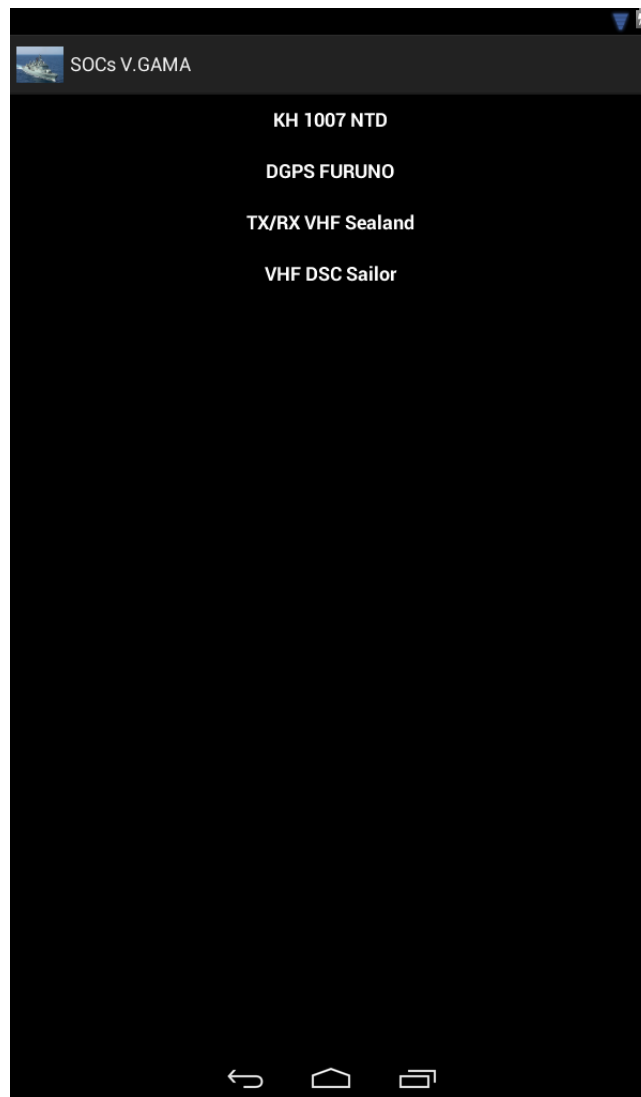
Em termos de funcionalidade, segue-se a descrição da aplicação, já com o *design* que se pretende para o protótipo final da mesma. Pretende-se que a aplicação a desenvolver tenha os seguintes *layouts* e características:

- Menu Principal: a figura 16 apresenta o menu com quatro botões. Cada botão identifica cada elemento da ponte que deve executar as SOC's.



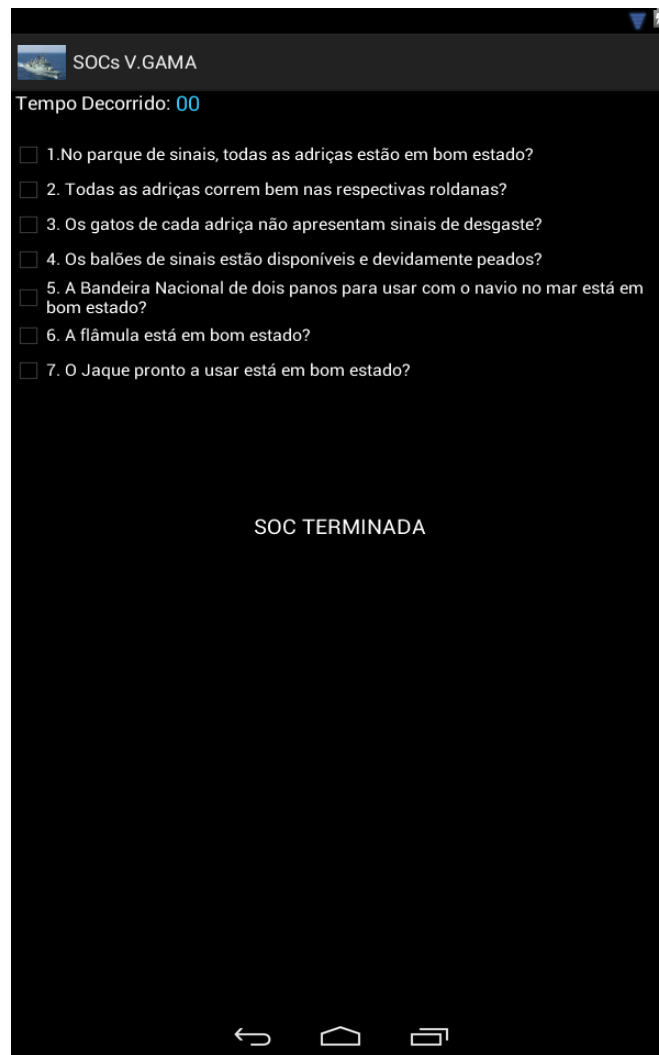
**Figura 16 - Menu Principal da aplicação móvel**

- Listagem dos equipamentos: ao seleccionar uma das opções do menu principal, surge a listagem dos equipamentos relacionado com o executante seleccionado.
- Corpo da SOC, ao escolher um dos equipamentos apresentados dependendo do executante, será apresentado a *checklist* do mesmo (ver figura 18). A estrutura das *checklists* é semelhante para todos equipamentos. O utilizador faz a escolha do item através de uma *checkbox*, onde caso seja efetuada a escolha de um dos itens, a aplicação entende registra que a verificação do item do equipamento foi feita com sucesso e o contrário representa uma resposta negativa.



**Figura 17 - Layout da Lista de Equipamentos do Oficial de quarto à ponte**

Depois de finalizada a verificação dos itens, o utilizador deve seleccionar o botão terminar, que tem como função parar o cronómetro e guardar o tempo decorrido (desde a seleção do equipamento até a seleção do botão terminar), e envia o relatório de estado de cada item da SOC.



**Figura 18 - Exemplo do Corpo da SOC**

Como a realização das SOC's no navio é feita com o objetivo de verificar a operacionalidade dos equipamentos, convém que o executante faça da melhor maneira. Deste modo, para a solução pretendida não foi implementado tempos de alerta para cada item da SOC. Porém, foi definido tempo mínimo padrão para execução da mesma (Sanches, 2012).

Durante o desenvolvimento da aplicação móvel foram efetuados testes de comunicações entre a aplicação móvel e o servidor, com o objetivo de detetar erros na transmissão de dados, ou mesmo erro de funcionamento da aplicação. De forma a garantir que na altura dos testes e validação a realizar com os utilizadores responsáveis pela execução das SOC's esses erros não surgissem.

### **4.3. Requisitos da Aplicação web**

No que concerne a aplicação web, pretende-se que a mesma disponibilize o acesso a base de dados aos outros elementos do navio, no intuito de acompanhar a execução das atividades das SOC's em tempo real; consultar os históricos de atividades já realizadas e também permitira fazer uma análise dos dados que serão disponibilizados.

Para definir os requisitos da aplicação, é necessário conhecer os objetivos que se querem ver cumpridos. Mas antes de definir os mesmos, convém salientar que a aplicação web deste projeto disponibilizará para os utilizadores as SOC's e as FOPs<sup>66</sup> do navio. Assim, e de acordo com a motivação que originou o desenvolvimento deste projeto, tem-se como principais objetivos da aplicação web os seguintes: permitir o acompanhamento em tempo real das tarefas por parte de outros elementos; analisar o desempenho dos elementos de quarto à ponte; possibilitar uma avaliação posterior das ações realizadas pelos elementos de quarto à ponte identificando os pontos críticos, ou seja, identificar quais as tarefas que não são realizadas ou que são realizadas fora do tempo definido; estabelecer o perfil de execução de cada uma das tarefas em função do histórico, da ordem pela qual são executadas e do tempo necessário para a sua execução; e, por fim recomendar as tarefas que requerem mais treino, com base naquelas onde habitualmente se falha ou se demora mais tempo do que o estipulado.

Seguidamente apresenta-se os principais requisitos da aplicação.

A aplicação web deverá permitir uma ligação à base de dados em tempo real de forma a disponibilizar a informação relativa a:

- Os elementos de quarto à ponte que executaram as SOC's;
- Registo do tempo em que cada SOC foi realizada, e se esse tempo está dentro do estipulado ou não;
- Os itens que falharam na execução da SOC;
- O tempo total da realização da SOC.

Por cada SOC deverá ser possível obter um relatório (exportável para PDF), com a indicação de todas as tarefas que devem ser realizadas, os respetivos tempos de referência, a indicação dos tempos médios de execução de cada tarefa, e ainda a indicação das tarefas que habitualmente não são realizadas.

---

<sup>66</sup> Fichas de Verificação do Oficial de Quarto-à-Ponte, estudo caso da dissertação da ASPOF EN-AEL Ganança do Carmo.

A página principal da aplicação (relativamente a SOC) que permite fazer a consulta em tempo real tem o *layout* apresentado na figura 19. O utilizador, de acordo com o equipamento na responsabilidade de cada elemento de quarto à ponte, pode consultar a execução da SOC ao seleccionar um deles.



**Figura 19 - Pagina principal das SOCs**

Após a seleção, é exibida uma página que mostra o estado atual do equipamento selecionado. Também é possível visualizar o tempo padrão de execução da SOC e o tempo que a mesma demorou para terminar. Caso a aplicação móvel não tenha enviado nenhuma atualização, na coluna do Estado é visualizado o valor *null*. Por outro lado, caso a aplicação já tenha atualizado, é possível ver o estado enviado pela mesma. Tal como foi referido na secção anterior, o estado de cada item é: “falhou” ou “Funciona”.



Itens da SOC	Estado
Para a total execução deste SOC, a consola a testar deve ser MASTER. (Se necessário premir "REQUEST MASTER") (1) No Display deverá assegurar as seguintes condições de modo a obter um bom desempenho no SOC:	null
a) GANHO entre 70 e 80;	null
b) SEA no mínimo nunca zero;	null
c) RAIN no mínimo.	null
(3) Selecciona a escala 12NM. No menu SELECT RADAR seleccionar TX MON VIDEO ON Aparece um "Sol" no centro do Radar cujo valor típico é 1NM e um "Pluma" com um comprimento típico de 6NM.	null
(4) Verifique que o Valor do HEADING lido no display (canto superior direito) é igual ao Repetidora da Giro.	null
(5) Verifique que o valor de STW (velocidade) no display (canto superior direito) é igual ao da repetidora de Odómetro.	null
(6) Compare a posição GPS indicada no canto inferior direito, com a posição do respectivo GPS seleccionado.	null

Duração da SOC:

Tempo padrão : 02:00

Data: 29/06/13

**Figura 20 - página de acompanhamento da SOC do Radar Kelvin Hughes<sup>67</sup>**

Como o sistema proposto também armazena o histórico das SOC's realizadas, foi também desenvolvido a página para consultar os dados guardados. A página de consulta apresenta numa tabela o histórico de todas as SOC's realizadas. Mostrando em três colunas distintas, os dados relativos a: duração da SOC, o equipamento e a data da realização, como é possível ver na figura abaixo.

Duração	Equipamentos	Data da Realização
3:13	VHF_Sealand	17/06/13
0:07	Adrica	17/06/13
0:57	Luzes	17/06/13
0:57	Luzes	17/06/13
0:40	Anemometro	17/06/13
0:40	Anemometro	17/06/13
null	null	null

Data:  exemplo: 18/06/2013 Equipamento:  (conforme a Tabela)

**Figura 21 - Pagina para a consulta do histórico**

<sup>67</sup> Como na data referida não tinha sido efetuada nenhuma SOC, o estado é *null*

#### 4.4. Estrutura do Código Fonte das Aplicações

A construção do sistema implementado nesta dissertação contempla várias tecnologias, umas específicas para o módulo do dispositivo móvel, outras específicas para o módulo Web, e outras comuns a ambos. Os dois módulos foram realizados na ferramenta de desenvolvimento Eclipse.

Para a comunicação entre o servidor de base dados e os dois módulos foi utilizada a tecnologia de *Web Services*. Esta tecnologia baseada em SOAP<sup>68</sup> permite a interação e colaboração entre os dispositivos da rede (W3C, 2005). O *Web Service* é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes, como é o caso desse projeto. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. Esta tecnologia permite às aplicações enviar e receber dados em formato XML. Cada aplicação pode ter a sua própria "linguagem", que é traduzida para uma linguagem universal, o formato XML (Moreira, 2005).

Utilizando a tecnologia *Web Service*, uma aplicação pode pedir a outra para executar tarefas mesmo que as duas aplicações estejam em diferentes sistemas e escritas em linguagens diferentes, como é o caso da presente dissertação.

Na programação dos módulos, para a aplicação móvel foi utilizada o Android (linguagem similar ao Java), já para a aplicação *web* foi utilizada a tecnologia JSP que tem incorporado a linguagem HTML para a criação das páginas. Para que a aplicação web pudesse aceder a base de dados foram utilizados *Scriptlets*<sup>69</sup>. Os Scriptlets servem de grande ajuda na programação das paginas em JSP, visto que permite escrever código em java numa pagina que suporta a linguagem HTML. Assim, foi possível aceder a base de dados, através da página JSP, sem criar classes auxiliares, como pode ser visto no Anexo C.

Quanto a base de dados, a mesma foi desenvolvida na linguagem SQL, foi utilizada a ferramenta de desenvolvimento MySQL Workbench, que permitiu criar as tabelas para o envio de dados e também para a consulta dos mesmos.

---

<sup>68</sup> Acrónimo para Simple Object Access Protocol, é um protocolo para troca de informações estruturadas em XML.

<sup>69</sup> Na tecnologia JSP, um scriptlet é um pedaço de código Java embutido em um código JSP semelhante a um código HTML.

O scriptlet sempre está dentro de tags `<% @ @%>`.

#### 4.4.1. Código fonte da aplicação móvel

Antes de avançar para a explicação de como está estruturado o código da aplicação móvel do sistema proposto na presente dissertação, é necessário compreender os conceitos básicos do sistema operativo Android e o modo como são criadas as aplicações.

Ao ser criado um novo projeto em android através do eclipse, são criados vários ficheiros no pacote *Explorer* no eclipse, tal como é possível verificar na figura 22.

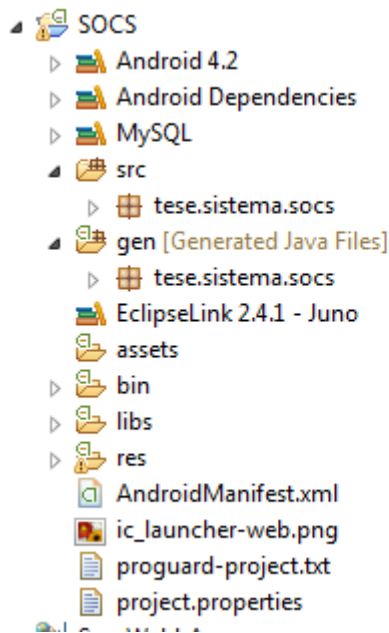


Figura 22 - Estrutura de uma aplicação android

Segue-se a descrição das pastas e os ficheiros que cada uma contém de acordo com Lee (2012, p. 29):

- *src* – contém os ficheiros de extensão *.java* da aplicação, nomeadamente as *activities* que definem o projeto da aplicação móvel. O código da aplicação é escrito dentro desta pasta, onde os ficheiros *.java* são listados dentro do pacote do *projecto* , que neste caso é *tese.sistema.socs*.
- *gen* – contém os ficheiros do tipo *R.java*, compilador responsável por todos os recursos (*resources*) da projeto. Não deve ser alterado, pois os recursos utilizados no desenvolvimento da aplicação são compilados automaticamente nesta classe.
- *Android 4.2* – Este item contém o ficheiro *android.jar*, onde podem ser encontrados todas as classes de bibliotecas necessárias para a aplicação móvel.
- *assets* – esta pasta contém todos os ficheiros ativos utilizados pela aplicação, tais como HTML, ficheiros de texto, base de dados, etc.

- *bin* – Nesta pasta é possível encontrar os ficheiros criados a partir do ADT, contendo todos os ficheiros e instruções necessárias para correr uma aplicação em android.
- *res*- Todos os recursos utilizados pela aplicação móvel encontram-se nesta pasta, são localizados os layouts utilizados na aplicação.
- *AndroidManifest.xml*- Neste ficheiro são especificadas as permissões necessárias para aplicação, tais como permissões de internet. Além disto, também são incluídas todas as atividades criadas afim das mesmas poderem correr no dispositivo móvel.

O ficheiro *AndroidManifest.xml* contém informações importantes para o funcionamento da aplicação, pois segundo Lee (2012, p. 31), o *AndroidManifest.xml* tem as seguintes funções:

- Define o nome do pacote;
- A versão do código da aplicação móvel;
- O nome da versão do android;
- A versão mínima do sistema operativo onde a aplicação irá ser instalada,
- A imagem atribuída a aplicação;
- O nome da aplicação móvel;
- As *activities* da aplicação;
- As permissões que o desenvolvedor queira implementar para a aplicação.

A descrição feita acima pode ser vista no Anexo C onde é possível observar o *AndroidManifest.xml* da aplicação móvel do sistema.

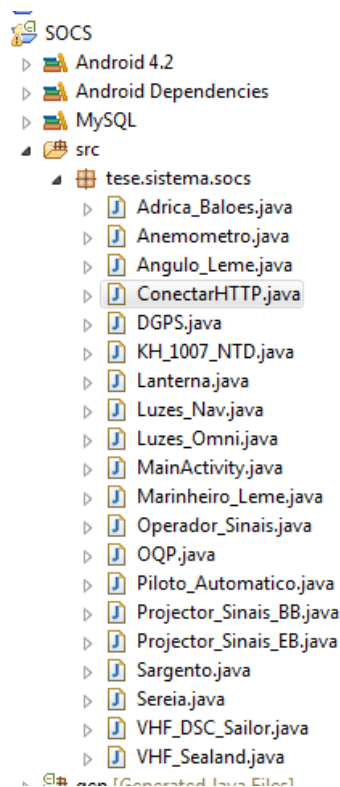
O desenvolvimento da aplicação foi elaborado de duas formas. Numa primeira fase foi desenvolvido o *layout* associado a cada *Activity*<sup>70</sup>. Deste modo, foram criadas no total 19 *activities*, organizadas da seguinte forma:

- *Activity* principal – é responsável pelo acesso as *activities* relacionadas com os elementos responsáveis pela realização das SOC's;
- *Activities* dos executantes – no total são quatro, onde cada uma delas permite o acesso as *activities* que estão relacionadas com os equipamentos de cada elemento;
- *Activities* dos Equipamentos – são catorze *activities*, onde cada uma permite o acesso a *checklist* de cada equipamento.

---

<sup>70</sup> Activity é uma janela que contém a interface do utilizador duma determinada aplicação. Definição traduzida de Lee( 2012, p. 29).

Além das *activities*, foi necessário criar uma classe em Java contendo os métodos com a função de conectar a aplicação a rede sem fio (trata-se da classe ConectarHTTP.java da figura 23). Cada vez que uma das *activities* relacionadas com os equipamentos é utilizada, é chamada a classe ConectarHTTP.java que no final de cada permitire fazer o envio dos dados através da rede sem fio.



**Figura 23 - Activities criadas para o desenvolvimento da aplicação móvel**

Durante a codificação da aplicação surgiu um pequeno “problema”, no que concerne ao envio dos dados para o SGBD. Pois as aplicações em Android não fazem uma ligação direta ao SGBD, no entanto, é possível enviar os dados utilizando *web services* (Lee, 2012). Deste modo, foi necessário criar os *web services*, aproveitando o facto de que a aplicação web foi desenvolvida utilizando a tecnologia JSP, a mesma também foi utilizada para os *web services*.

Foram criados *web services* para cada equipamento, com o objetivo de enviar os dados para o SGBD (ver figura 24).

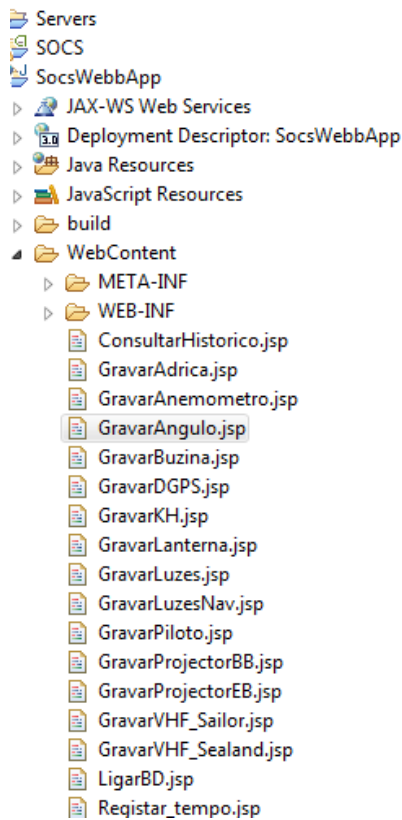


Figura 24 - Lista dos *Web Services* utilizados no desenvolvimento da aplicação

#### 4.4.2. Código fonte da aplicação Web

Depois de ter sido feito o enquadramento tecnológico JSP no estudo do estado da arte do projeto, no que concerne as tecnologias web, serão agora descritos os conceitos e sintaxe básica da tecnologia JSP.

O código JSP é inserido nos próprios ficheiros HTML, estando as instruções misturadas com as marcas HTML. A identificação das zonas que são JSP e das que são HTML é feita pelo interpretador Java, que identifica as partes a interpretar. Os ficheiros com JSP têm a extensão. jsp, necessária para que o interpretador saiba em que condições interpretar o conteúdo (Coelho P. , 2005).

De acordo com Coelho (2005), as instruções em JSP podem dividir-se em três elementos:

- Diretivas;
- Elementos de *Scripting*;
- Ações.

## Diretivas

As diretivas são mensagens que são dirigidas ao motor JSP, dando instruções ao mesmo de como processar o código. Definem-se sempre com a seguinte sintaxe: `<%@...conteúdo...%>`.

Nas páginas desenvolvidas nesta dissertação, foram utilizadas as duas diretivas que se seguem: `<%@ page contentType="text/html" %>` e `<%@ page import = "java.util.Date" %>`. Ambas são consideradas diretivas de página, onde definem e manipulam atributos que afetam toda a página JSP. Nestes exemplos acima referido estamos a dizer ao motor JSP que o tipo de conteúdo (MIME) a ser gerado é HTML e é necessário importar o pacote java `"java.util.Date"` (Coelho P. , 2005). No desenvolvimento da aplicação web foram utilizados atributos das diretivas de página donde se destacam as seguintes:

- ✓ **language** - Linguagem de *scripting* a ser utilizada (normalmente java);
- ✓ **import** – Lista de pacotes Java a importar;
- ✓ **session** – Define se a página está envolvida numa sessão http (*true* ou *false*);
- ✓ **contentType** – Tipo MIME da resposta.

## Elementos *Scripting*

Nesta categoria de elementos enfiavam-se as mais importantes características de sintaxe das JSP. Trata-se das múltiplas instruções que implementam a lógica de programação diversa, tal como a declaração de variáveis, ciclos, condicionais, etc. Estes elementos são parte fundamental da sintaxe básica, incorporados em *scriptlets*. Os *scriptlets* incorporam código java sob forma de *scripts* interpretados, constituído o verdadeiro motor dinâmico das JSP.

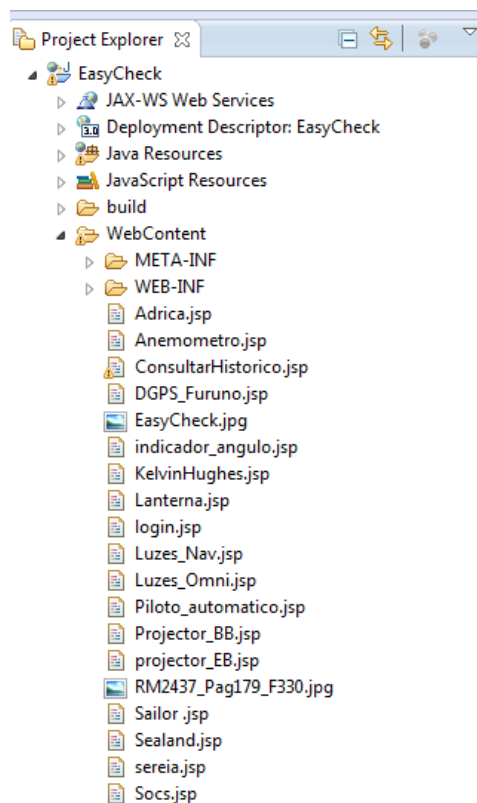
## Ações

As ações são instruções especiais, marcas que afetam o comportamento do motor de busca, influenciando a resposta a dar ao cliente final. Não são parte da sintaxe lógica do programa, mas antes consistem em comandos que ordenam interações com componentes, atos de visualização, tais como redirecionamento ou inclusão de código (Coelho P. , 2005).

O desenvolvimento da aplicação web envolveu uma maior interatividade com a base de dados, visto que a mesma ia servir de consulta dos eventos a que estivessem a decorrer e

também para consultar os eventos anteriores. Assim o desenvolvimento da mesma foi feito em duas partes:

- **Atividade a decorrer** – apresenta uma página inicial com os elementos de responsáveis pela execução, onde cada tem dá acesso a um menu com os equipamentos, permitindo assim acompanhar a atividade em tempo real. Foram criadas páginas de acordo com os equipamentos alvo da presente dissertação, tal como mostra a figura 25.
- **Atividade para consultar Histórico** – esta página foi criada para permitir o utilizador consultar atividades anteriores, efetuando a pesquisa por data e equipamento referente a data.



**Figura 25 - Páginas web criadas para aplicação Web**

#### **4.4.3. Estrutura da Base de Dados**

Como foi referido na secção anterior, a aplicação web interage muito com a base de dados, sendo a interação uma das características mais importantes de qualquer aplicação



web (Diogo & Graça , 2011). A base de dados funciona como arquivo de informação a ser incorporado de forma dinâmica nas páginas web que a elas se ligam.

Para efetuar a ligação entre uma aplicação java e um determinada base de dados, que neste caso trata-se do MySQL, é necessário instalar um *driver* específico que faça a ligação entre o JDBC<sup>71</sup> e o MySQL.

Segundo Coelho P. (2005), para utilizar o JDBC, há um conjunto de passos que têm de ser seguidos, nomeadamente:

- Carregar o driver através da instrução, `Class.forName`, para que o *software* Java possa comunicar com o SGBD;
- Definir o URL de ligação, que define o local onde está acessível a base de dados;
- Criar a conexão através da utilização de um objecto de tipo “*Connection*”, onde é passado o URL, o *user* e a *password* de acesso;
- Criar um objeto *statement*, que é utilizado para depois executar instruções;
- Executar o pedido (*query*) ou alteração (*update*) a partir do objeto *statement*, eventualmente obtendo os resultados num objecto *ResultSet*;
- Obter e tratar dos resultados que eventualmente estejam no *ResultSet*;
- Fechar a conexão.

É possível observar as instruções acima citadas sendo utilizadas na criação da *web service* para enviar as informações à base de dados.

```
<%  
java.sql.Connection conn = null;  
String url = "jdbc:mysql://localhost/test?user=root&password=tese";  
java.sql.Statement st;  
java.sql.ResultSet rs;  
  
String guarda; // variavel que vai guardar as alterações na base de dados  
String guarda_data;  
String Data;  
Data = (new Date()).toString();  
  
Data = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new Date())).toString();  
Class.forName("com.mysql.jdbc.Driver").newInstance();  
conn = DriverManager.getConnection(url);  
  
st = conn.createStatement();  
guarda = "insert into adrica_baloes (Estado,Data) values" + "(" +  
request.getParameter("Estado")+"", (" "+Data+"") );  
  
st.executeUpdate(guarda);  
conn.close();  
%>
```

Para criar a base de dados, primeiramente, foi necessário definir que tipo de informação ia conter cada tabela. Assim, o conteúdo de cada tabela tinha de estar de acordo com o tipo de dados enviados pela aplicação. Assim, foram definidas duas variáveis para a tabela de cada equipamento, onde uma ia conter os dados relativos ao estado do equipamento e a outra com a data. Porém, conforme definido, a base de dados também servirá para consulta. Deste modo, foi necessário criar uma coluna de identificação, que permitira aceder ao conteúdo a ser consultado através da página web.

A codificação da base de dados foi feita através de tabelas. Foram criadas o número de tabelas proporcional ao número de equipamentos utilizados na aplicação móvel.

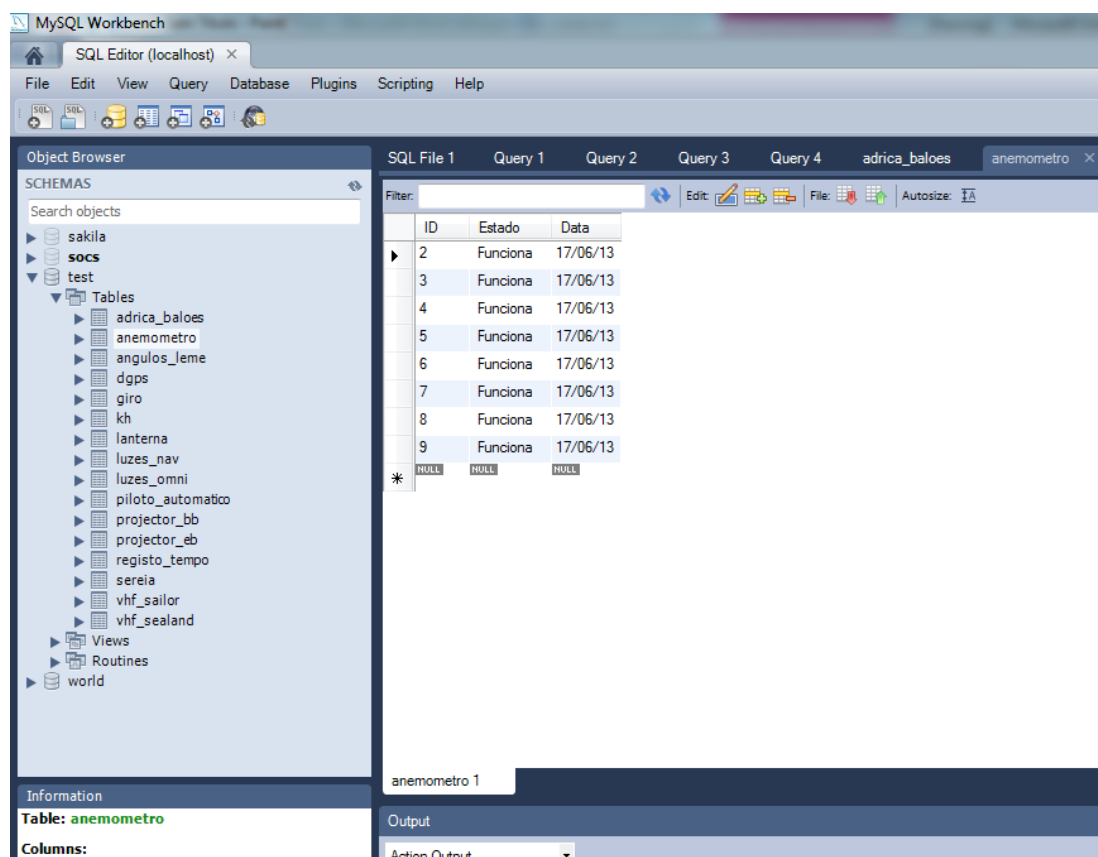
Cada tabela dos equipamentos tem três colunas: ID, Estado e Data.

- A coluna *ID* contém os números criados automaticamente pela base de dados (utilizando a opção *AI-Auto Increment*), serve de identificação da linha da tabela.
- Na coluna *Estado* é guardado o estado dos equipamentos (falhou ou funciona), enviado pela aplicação móvel;
- Na coluna *Data* é armazenado a data da realização da SOC, introduzido através do *web service* utilizado entre a aplicação móvel e a base de dados.

A figura 26 apresenta as tabelas que constituem a base de dados, e as informações arquivadas dum dos equipamentos<sup>72</sup>.

---

<sup>72</sup> Dados referentes ao teste realizado no SIMNAV da EN no dia 17 de Junho de 2013



**Figura 26 - Tabelas da Base de Dados**

Além das tabelas relacionadas com os equipamentos e o respetivo estado de cada item efetuado, também foi implementada uma tabela para registar a duração de cada SOC. Tal como foi definido nos requisitos da aplicação web, pretende-se que o sistema proposto permita consultar o histórico do tempo de duração das SOC's. Desta forma, para que fosse possível efetuar o registo da duração e o equipamento associado ao tempo decorrido, foi criada a tabela denominada *registro\_tempo*. A tabela contém três colunas. Além da coluna *ID* e a coluna *Data*, semelhantes às colunas dos equipamentos, apresenta duas novas colunas:

- A coluna *Duracao* contém o tempo de realização de cada uma das SOC, enviado pela aplicação móvel;
- A coluna *Equipamento* regista o equipamento relacionado com o tempo enviado pela aplicação móvel;

ID	Duracao	Data	Equipamento
28	3:13	17/06/13	VHF_Sealand
29	0:07	17/06/13	Adrica
30	0:57	17/06/13	Luzes
31	0:57	17/06/13	Luzes
32	0:40	17/06/13	Anemometro
33	0:40	17/06/13	Anemometro

**Figura 27 - Tabela com o Registo dos tempos realizados em cada SOC<sup>73</sup>**

---

<sup>73</sup> Dados referentes ao teste realizado no SIMNAV da EN no dia 17 de Junho de 2013.

## CAPITULO V TESTES E VALIDAÇÃO

### 5.1. Objetivos e condições dos Testes

A fase de testes e validação do sistema proposto possibilita verificar se a aplicação final corresponde aos objetivos traçados. Se funciona corretamente nas plataformas a que se destina e se vai de encontro às necessidades do utilizador final. Para isso foram realizados testes de forma a manter uma lista detalhada de erros encontrados e da sua severidade, de modo a que estes possam ser posteriormente corrigidos.

De acordo com a constituição do sistema proposto, para a realização dos testes foi necessário garantir os seguintes elementos:

- Servidor - foi utilizado como servidor um computador portátil com o Toshiba Qosmio, com 8Gbytes de memória RAM, Intel® Core™2 i7-2630QM (2 GHz) a correr o Windows 7 Ultimate na sua versão de 64 bits.
- Um cliente fixo - para o cliente fixo foi utilizado o mesmo computador utilizado como servidor;
- Um dispositivo móvel, onde a aplicação móvel foi instalada num *tablet* Asus Google Nexus 7" com sistema operativo Android 4.2, processador NVIDIA® Tegra® 3 quad-core 1,2 GHz e ecrã com resolução 1280x800 pixels com conectividade a rede sem fio Wi-Fi usando o protocolo IEEE 802.11g/b
- Ponto de acesso- como ponto de acesso foi utilizado um dispositivo móvel (iPad) que serviu de *hotspot*<sup>74</sup> para conectar o servidor e o dispositivo móvel a uma rede sem fio.

A equipa de testes deve ser composta, pelo mestrando que desenvolveu as aplicações, e por mais três elementos que desempenhem as funções de Oficial de Quarto à Ponte, Sargento de Quarto à Ponte e Marinheiro de Leme. É importante que os três elementos não tenham participado no desenvolvimento das aplicações, para que os resultados dos testes não sejam tendenciosos. Cada elemento utiliza um modelo de relatório de teste que contém os requisitos propostos para serem seguidos durante a execução dos casos de teste. Este relatório terá ainda um pequeno questionário para avaliar o grau de satisfação do utilizador relativamente às aplicações. O mestrando terá ainda um relatório para preencher, onde deve registar os tempos que os utilizadores levaram a

---

<sup>74</sup> *Hotspot* é o nome dado aos locais públicos onde há cobertura do serviço de comunicações, normalmente internet, sem fio através da tecnologia Wi-Fi.

executar as várias atividades com a ajuda das *checklists* em papel e da aplicação móvel, para posterior comparação dos resultados. Quanto a aplicação web, a consulta dos dados será feita pelo mestrando. O autor deve chamar os elementos participantes no teste a fim de mostrar na aplicação web o resultado das atividades executadas através da aplicação móvel, onde os mesmos poderão ver se o tempo decorrido para cada ação e também um relatório de todas as SOC's realizadas durante o teste.

Os testes são realizados no simulador de navegação (SIMNAV), a execução das aplicações será feita em movimento usando um dispositivo móvel ligado à rede Wi-Fi do computador que será o servidor. Posteriormente serão realizados os mesmos testes a bordo de um navio da Marinha Portuguesa.

As aplicações são testadas da seguinte forma: cada elemento realiza as tarefas das SOC's que lhes compete de forma tradicional, auxiliados por um documento em papel com os itens a verificar. Depois serão realizados os mesmos testes com a ajuda do dispositivo móvel. É realizado um relatório de teste onde será comparado o desempenho, eficácia, eficiência e opinião pessoal dos utilizadores quanto à realização das atividades utilizando papel e o dispositivo. Desta forma, cada elemento preenche um relatório referente aos testes das SOC's para análise.

Os objetivos dos testes e validação consistem em comprovar a viabilidade do uso das novas tecnologias para redução do excesso de papel existente a bordo dos navios, bem como melhorar o desempenho dos utilizadores na realização das diversas atividades devido à interatividade que a aplicação móvel apresenta. Esta situação é impossível de obter em documentos em papel. Estes fatores aliados à capacidade de registar a informação de interesse após terminada a atividade, são mais-valias evidentes. Contudo, não descartam a necessidade de realização de testes que verifiquem e validem as melhorias e vantagens que esta nova abordagem traz.

Assim, durante a realização dos testes para a aplicação móvel deverão ser efetuados os procedimentos descritos na tabela 8 (e também no descritos no Anexo F):

<b>Identificação</b>	T01
<b>Descrição</b>	Teste SOC OQP – equipamentos na responsabilidade do oficial de quarto à ponte.
<b>Pré-Condições</b>	Menu principal no ecrã do dispositivo; Registo de tempo padrão de realização de cada equipamento; Realizar a SOC de forma tradicional, em papel.
<b>Ação a Ser Tomada</b>	Realizar a SOC, validar os itens da aplicação ao executar as respetivas atividades.
<b>Resultado Esperado</b>	A aplicação deverá registar os tempos em que cada SOC foi realizada, ao terminar a SOC a aplicação envia a informação para o servidor com os tempos de cada SOC;  A aplicação deverá enviar para o servidor uma informação de estado de cada item da SOC (Falhou, ou Funciona Corretamente);  A Aplicação deverá enviar para servidor o nome do equipamento e a data da realização da SOC.
<b>Método para Validação</b>	O registo dos tempos está a funcionar corretamente; Aplicação estabelece ligação com servidor e envia os dados; O tempo decorrido é inferior ou igual ao tempo padrão.

**Tabela 8 - Procedimentos de Teste da Aplicação Móvel**

Quanto a aplicação web, também deverão ser efetuados os procedimentos descritos na tabela abaixo:

<b>Identificação</b>	T04
<b>Descrição</b>	Teste SOC Equipamentos – Consulta do estado dos equipamentos.
<b>Pré-Condições</b>	Página principal no ecrã do computador; Registo de tempo padrão de realização de cada equipamento; Realizar a SOC de forma tradicional, em papel.
<b>Ação a Ser Tomada</b>	Fazer a consulta de todas as páginas, cujos equipamentos foram testados na aplicação móvel
<b>Resultado Esperado</b>	A aplicação deverá ser capaz de consultar os dados enviados da aplicação móvel ao servidor.
<b>Método para Validação</b>	Os dados enviados condizem com o que foi efetuado na aplicação móvel; Aplicação estabelece ligação com servidor e consulta os dados; É possível fazer a consulta do histórico.

**Tabela 9 - Procedimento de Teste da Aplicação Web**

## 5.2. Descrição dos testes realizados

Os testes foram realizados no SIMNAV<sup>75</sup> da Escola Naval e no NRP “João Coutinho”<sup>76</sup>.

No que concerne ao teste efetuado no SIMNAV, foram testados os equipamentos atribuídos ao Sargento de quarto à ponte e ao Marinheiro do Leme. Não foi possível testar os outros equipamentos, visto que a maioria dos seus equipamentos só existem nos navios. Participaram nos testes três elementos pertencentes ao serviço de navegação da Escola Naval que seguiram as recomendações dadas pelo mestrando.

Já o teste realizado no navio permitiu efetuar a experimentação de mais equipamentos<sup>77</sup>. Para que o mesmo fosse possível, foram disponibilizados ao mestrando três elementos da guarnição e o oficial do serviço de operações do navio.

Ambos os testes consistiram na realização das SOC's dos equipamentos em questão, utilizando os dois métodos, em papel e o dispositivo móvel conforme foi definido na secção anterior. Foram registrados os tempos de execução de cada um dos métodos, e foram comparados os resultados. Além disto, foi possível consultar em tempo real o estado de execução dos testes através da aplicação web.

Os testes foram subdivididos em series, conforme se segue:

- **Teste no SIMNAV** – quatro séries de testes, onde a primeira foi realizada utilizando o método tradicional, e foi definido o tempo padrão para o término da SOC de cada equipamento. As restantes foram efetuadas utilizando o dispositivo móvel;
- **Teste no Navio** – seis séries de testes, as duas primeiras também foram efetuadas pelo método tradicional e as restantes através da aplicação móvel, tal como no SIMNAV.

## 5.3. Resultados obtidos

Na realização dos testes no SIMNAV da Escola Naval, primeiramente foi efetuado uma explicação sobre o sistema e os objetivos dos testes aos elementos participantes. De seguida foi definido o tempo padrão para execução das SOC's de cada equipamento (Bué, 2013). De acordo com Bué (2013), os tempos utilizados como referência são os que se

---

<sup>75</sup> Acrónimo para Simulador Naval

<sup>76</sup> Navio onde o Mestrando efetuou o estágio de embarque

<sup>77</sup> Exceto o KH 1007 NTD.



encontram descritos na tabela 10. Deste modo, para o sistema proposto na presente dissertação, o tempo padrão de cada equipamento passa a ser o indicado na tabela.

<b>Equipamentos</b>	<b>Tempo Padrão (em minutos)</b>
Indicador de Angulo do Leme	5:00
Anemómetro	1:00
Sereia	1:30
Luzes de Navegação	1:00
KH 1007 NTD	2:00
VHF DSC Sailor	5:00
VHF Sealand	4:00
DGPS FURUNO	2:00
Projector de EB	1:00
Projector de BB	1:00
Adriças e Balões	2:30
Lanterna de Sinal Portátil	1:00
Luzes Omnidirecionais	1:30

**Tabela 10 - Tabela do tempo padrão definido no SIMNAV<sup>78</sup>**

Após a definição do tempo padrão para cada equipamento, foram criadas as condições para serem efetuados os testes. Onde iam ser realizadas as quatro séries descritas na secção anterior. Convém salientar que foi dado cinco minutos para cada elemento adaptar-se a aplicação móvel, e de seguida exporem as dúvidas relativamente a mesma. Os elementos participantes utilizaram apenas a aplicação móvel, a consulta dos dados através da aplicação web foi feita pelo mestrando, conforme definido.

Quanto ao teste realizado no navio, primeiramente foi dada a explicação do sistema a testar, e foi feito o esclarecimento de dúvidas apresentadas pelos elementos que faziam parte da equipa de testes. Seguidamente foram criadas as condições para realização dos testes, dando tempo para adaptação à aplicação móvel. Posteriormente foram realizadas as seis séries definidas para o teste no navio. É importante frisar que no navio foi possível realizar SOC's de mais equipamentos, permitindo assim uma melhor avaliação dos resultados obtidos. Os resultados dos testes e a análise aos mesmos serão feitos nas secções 5.3.1 e 5.3.2.

---

<sup>78</sup> Tabela efetuada de acordo com a informação prestada pelo oficial do serviço de navegação da Escola Naval 1TEN Isabel Bué

### 5.3.1. Desempenho

A otimização dum sistema já existente numa organização envolve um conjunto de fatores a serem analisados, como por exemplo se o projeto irá ou não beneficiar os utilizadores interessados. No caso das SOC's, por se tratar de atividades realizadas nos navios da Marinha Portuguesa, a solução proposta não deve baixar os padrões operacionalidade dos mesmos. Sendo assim, para avaliar o sistema proposto na presente dissertação foram realizados os testes já descritos, cujos resultados são apresentados nas tabelas que se seguem (tabela 11, tabela 12 e tabela 13).

	<b>Série 1 (papel)</b>	<b>Série 2 (dispositivo)</b>	<b>Série 3 (dispositivo)</b>	<b>Série 4 (dispositivo)</b>	<b>Tempo Padrão</b>
Ind. de Angulo do Leme	2:00	2:03 min	2:00 min	1:54 min	5:00 min
Anemómetro	31 seg	40 seg	40 seg	41 seg	1:00 min
Sereia	55 seg	58 seg	53 seg	51 seg	1:30 min
Luzes de Navegação	29,5 seg	57 seg	43 seg	34 seg	1:00 min
KH 1007	1:00 min	1:30 min	1:25 min	1:23 min	2:00 min

**Tabela 11 – Comparação do registo de tempo efetuado no SIMNAV com o tempo padrão**

	<b>Série 1 (papel)</b>	<b>Série 2 (papel)</b>	<b>Série 3 (dispositivo)</b>	<b>Série 4 (dispositivo)</b>	<b>Série 5 (dispositivo)</b>	<b>Série 6 (dispositivo)</b>	<b>Tempo Padrão</b>
Ind. de Angulo do Leme	3:00 min	2:00 min	1:23 min	1:54 min	1:51 min	1:50 min	5:00 min
Anemómetro	35 seg	30 seg	30 seg	41 seg	42 seg	41 seg	1:00 min
Sereia	50 seg	48 seg	49 seg	51 seg	53 seg	52 seg	1:30 min
Luzes de Navegação	30 seg	37 seg	39 seg	34 seg	32 seg	31 seg	1:00 min
Piloto Automático	1:50 min	1:50 min	1:21 min	1:23 min	1:20 min	1:16 min	2:30 min

**Tabela 12 - Comparação do registo de tempo efetuado no Navio com o tempo padrão**

	<b>Série 1 (papel)</b>	<b>Série 2 (papel)</b>	<b>Série 3 (dispositivo)</b>	<b>Série 4 (dispositivo)</b>	<b>Série 5 (dispositivo)</b>	<b>Série 6 (dispositivo)</b>	<b>Tempo Padrão</b>
Luzes omnidirecionais	1:00 min	1:00 min	1:00 min	57 seg	57 seg	50 seg	1:30 min
Lanterna de sinal Portátil	49 seg	46 seg	47 seg	46 seg	48 seg	46 seg	1:00 min
VHF Sealand	3:00 min	3:30 min	3:30 min	3:02 min	2:57 min	2:58 min	4:00 min
DGPS Furuno	1:20 min	1:22 min	1:00 min	1:05 min	1:02 min	1:00 min	2:00 min
Adriças e Balões	1:53 min	1:53 min	1:21 min	1:23 min	1:20 min	1:16 min	2:30 min
Projetor de EB	59 seg	56 seg	57 seg	54 seg	54seg	55 seg	1:00 min

**Tabela 13 - Comparação do registo de tempo efetuado no Navio com o tempo padrão (continuação)**

Analisando as tabelas 11, 12 e 13 nota-se que o dispositivo móvel não influencia negativamente o desempenho dos utilizadores na execução das SOC's, apesar de em alguns casos, o tempo de execução no método tradicional apresentar melhores resultados que o do dispositivo móvel, devendo-se ao fator adaptação do novo sistema. Como o novo sistema não afeta o tempo que normalmente é executado cada SOC, é possível evidenciar as vantagens que o mesmo oferece em relação ao método tradicional.

Ora, os tempos registados nas tabelas foram observados, após o término de cada SOC, através da aplicação web. Esta observação não era possível ser feita no método em papel. Pois no método em papel os elementos responsáveis por relatar o estado dos equipamentos ao comandante do navio têm a informação das SOC's apenas quando todas se encontram realizadas.

O novo sistema permite a quem de direito acompanhar as SOC's, e de acordo com esta observação prever o atraso no aprontamento do navio, caso a execução de algumas das SOC's ultrapasse o tempo padrão, pois é possível visualizar através da aplicação web, se as SOC's decorrem na normalidade ou não. E desta forma é possível intervir a tempo, invés de fazer apenas quando todas estejam realizadas. Assim, é possível obter um melhor desempenho na elaboração das SOC's, principalmente no que concerne ao controlo, visto que as mesmas podem ser acompanhadas por outros elementos para além dos executantes.

Também é possível diminuir o número de arquivos em papel relativamente aos registos das SOC's, pois o novo sistema guarda os dados em formato digital na base de dados, como foi possível observar no teste realizado a bordo. Aproveitando o fato de já existir um histórico relativo ao teste realizado no SIMNAV da Escola Naval, foi possível fazer a consulta de histórico por data de realização e por equipamentos. Permitindo desta

forma, testar a aplicação web no que concerne a consulta de histórico. De acordo com os oficiais que participaram nos testes, o sistema proposto poderá servir de ajuda aos briefings ao comando, visto que apresenta um relatório sobre o estado de cada item das SOC's de cada equipamento, e também pela primeira vez é possível saber quanto tempo decorreu a execução de cada. Segue-se um quadro comparativo entre os dois sistemas, permitindo assim perceber quais as vantagens e desvantagens que o novo sistema apresenta em relação ao método atual.

	<b>Vantagens</b>	<b>Desvantagens</b>
<b>Sistema em Papel</b>	Existência de <i>dossiers</i> com o registo das SOC's	Não tem o registo do tempo decorrido na execução de cada SOC
	Em termos de tempo de execução apresentam resultados semelhantes ao sistema informático	Não é possível fazer o acompanhamento da SOC, ou saber quais já foram feitas.
	As SOC's são mais detalhadas	Não está estipulado tempo para terminar a SOC dum determinado equipamento
	Independência da rede sem fio;	A informação sobre o estado da SOC's só é feita após terminarem todas elas.
<b>Sistema informático</b>	Registo do tempo de execução de cada SOC	Os utilizadores ainda não estão adaptados
	Acompanhamento de cada SOC, assim que cada uma delas termina.	Em relação aos itens de cada SOC, apenas enviam a informação falhou ou funciona. Não especifica a falha ou o estado do equipamento em si.
	Partilha rápida da informação das SOC's na rede;	Caso não haja rede sem fio é impossível utilizar este sistema;
	Diminuição dos arquivos em papel	Caso o servidor esteja <i>offline</i> , não é possível fazer a consulta do histórico.

### 5.3.2. Viabilidade

De acordo com Vasconcelos (2013), o estudo de viabilidade pode ser classificado da seguinte forma:

- **Viabilidade operacional** - é uma medida do grau de adaptação do projeto para a organização. É também uma avaliação de como as pessoas se sentem em relação ao novo sistema/projeto.
- **Viabilidade técnica** - é uma avaliação da aplicabilidade de uma solução técnica específica e a disponibilidade dos recursos técnicos e dos especialistas.
- **Viabilidade de cronograma** - é uma avaliação de quão razoável está o cronograma do projeto.
- **Viabilidade econômica** - é uma avaliação de custo-eficiência de um projeto ou solução. Conhecida como análise de custo-benefício.

De acordo com os tipos de viabilidades apresentados, para a presente dissertação foi efetuado uma breve análise a viabilidade operacional do sistema. Esta análise foi feita com a intenção de obter a opinião dos utilizadores acerca do sistema a implementar.

Durante as fases de investigação do projeto até a elaboração do produto final foram feitas entrevistas a oficiais da Marinha Portuguesa, no intuito de perceber se a solução a implementar pode ser uma mais-valia ou não para organização. A quando da realização dos testes, o sistema apresentado despertou a curiosidade dos utilizadores, visto que nunca tinham efetuado as SOC's através de um dispositivo móvel. Os mesmos afirmaram que utilização de tecnologias de informação na Marinha é uma mais-valia, desde que sejam salvaguardados a confidencialidade, pois estamos perante uma organização militar.

Durante a realização dos testes, como já referido na secção 5.2, foi dado tempo para adaptação a aplicação móvel, e durante este período também surgiram dúvidas relacionados com a aplicação. Visto que os utilizadores estavam acostumados a efetuar as SOC's com três opções (Operacional, Inoperacional e degradado), conforme é possível ver no Anexo B, além disto, tinham a opção de colocar observações que permitia explicar a razão de um item não ter funcionado.

Quanto ao dispositivo móvel escolhido para realização dos testes, de acordo com os utilizadores, foi uma boa escolha. Visto que em termos de tamanho e de facilidade de manuseamento os *mini-tablets* são adequados para a situação. Porém, o facto de haver

zonas com pouca intensidade de sinal de rede, a vantagem de ter um dispositivo móvel seria melhor caso existisse vários pontos de acesso.

#### 5.4. Limitações e Problemas encontrados

Durante a realização dos testes ao sistema foram detetados erros nas aplicações e também falhas inerentes aos dispositivos móveis. Na tabela 15 é resumido os erros e limitações encontradas durante os testes do sistema proposto.

Tipo de erros e Limitações	Descrição
Falha na deteção de rede sem fio	Durante a realização dos testes, quer no SIMNAV quer no Navio, houve falha no envio de dados para o servidor devido a falha de rede. Tendo em conta que só havia um ponto de acesso.
Falha na execução da aplicação	Também foi detetada uma falha na execução da aplicação móvel, particularmente num dos equipamentos a cargo do operador de sinas (projetor de Sinais de BB). O item Projetor de Sinais EB quando era selecionado o programa terminava.
Falha no envio de dados	Quanto a realização da SOC da Sereia, equipamento na responsabilidade do sargento de quarto à ponte, não foi possível enviar os dados numa primeira tentativa, pois o ip definido no <i>Web Service</i> não era o ip do computador servidor. No entanto, foi possível resolver o problema rapidamente.
Falha na pesquisa dos dados	A pesquisa do histórico das SOC's é feita por data e por equipamento, porém para certos equipamentos não era possível fazer a pesquisa, devido a erro de programação. Corregido a posterior.
Impossibilidade de implementação do sistema na rede interna do navio	No teste a bordo, não foi possível colocar o sistema a funcionar na rede interna do navio, no entanto só foi possível fazer utilizando os equipamentos descritos na secção 5.1.

**Tabela 15 - Erros e Limitações encontrados durante os testes**

## CAPITULO VI CONCLUSÕES

### 6.1. Conclusão

Nesta dissertação de mestrado foi proposto a elaboração de um sistema que permitisse o auxílio na execução das SOC's através dum dispositivo móvel. O sistema também devia permitir a consulta em tempo real das atividades em execução, e a possibilidade de consultar o histórico das mesmas. Assim, durante este trabalho, verificou-se o estado da arte no que concerne aos seguintes temas: dispositivos móveis, sistemas gestores de base de dados, tecnologia de desenvolvimento de páginas web, e finalmente os serviços de comunicações em rede (*ethernet* e *wireless*). Esta verificação permitiu definir a tecnologia adequada para o sistema.

Por forma a otimizar a forma como as SOC's são realizadas foram desenvolvidos nesta dissertação duas aplicações, aplicação móvel e aplicação web. Onde a aplicação móvel auxilia o elemento responsável na execução das SOC's, e a aplicação web permite o acompanhamento das atividades em tempo real e também do histórico.

Com o sistema desenvolvido foi possível atingir os seguintes objetivos:

- Disponibilizar na rede a informação associada as SOC's, permitindo assim a partilha de informação em tempo real para os elementos do navio;
- Criação de histórico em formato digital, que possibilitará a diminuição dos volumosos *dossiers* em papel existentes no navio;
- Disponibilizar o tempo de execução da SOC de cada equipamento, em vez de ter apenas o tempo total de todas.

De forma a obter uma melhor perceção do funcionamento do sistema proposto, foram efetuados testes, e assim foi possível obter a opinião dos utilizadores em relação a o novo modo de execução das SOC's. Como são atividades com baixo níveis de *stress*, o dispositivo não foi um causador de perturbação na realização das tarefas para os utilizadores, pois foi fácil a adaptação ao mesmo, daí que em nenhum dos testes realizados foi obtido um tempo abaixo do padrão definido, cumprindo assim com os objetivos traçados. No entanto, como foi visto no capítulo anterior (tabelas 11, 12 e 13), os tempos registados através da aplicação móvel em algumas situações têm valores superiores e outras vezes inferiores quando comparado com o método tradicional, devendo-se a adaptação do utilizador a aplicação, pois um simples atraso ao clicar o botão enviar pode

resultar numa diferença de segundos. Diferença esta que não influencia o aprontamento do navio, pois os mesmos nunca chegaram a ultrapassar o tempo padrão.

## **6.2. Recomendações para trabalhos futuros**

Apesar dos objetivos terem sido alcançados, durante o desenvolvimento do projeto foram detetados elementos que podem vir a ser melhorados, ou em certos casos, podem vir a ser estudados. Ora, as sugestões vão ser abordadas em três partes distintas: aplicação móvel, aplicação web e Testes.

**Aplicação Móvel:** a aplicação pode ser melhorada se ao invés de ser tudo programado por *activities*, utilizar *fragments*. Ou seja, para cada executante atribui-se *fragments* para os equipamentos. Visto que os mesmos permitem colocar no mesmo ecrã os equipamentos todos, e o utilizador só tem de seleccionar o separador correspondente ao equipamento desejado. Porém não foi possível implementar neste trabalho porque sempre que houvesse mudança de separador, o contador reiniciava; Outro aspeto a ser melhorado na aplicação móvel é o facto de a mesma só poder ser utilizada com o dispositivo na posição vertical, pois quando era alterado para a posição horizontal (ou da horizontal para vertical) o contador também reiniciava; Ainda na aplicação móvel, também sugeria que fosse acrescentado mais equipamentos.

**Aplicação Web:** apesar de ter trabalhado com a ASPOF EN-AEL Gança do Carmo com o objetivo de ter apenas uma aplicação web com duas opções, uma para as FOPs e outra para as SOC's, não foi possível desenvolver uma mesma aplicação para ambos. Assim, caso alguém venha dar continuidade a este projeto, seria bom que as atividades (FOPs e SOC's) estivessem na mesma aplicação web.

**Testes:** Quanto aos testes, apesar dos mesmos terem corrido bem, não foi possível efetuar o teste utilizando mesmo a rede interna do navio, mas sim com adaptações, conforme foi descrito na secção 5.4, deste modo, sugere-se que, caso possível, sejam efetuados testes utilizando a rede do navio.



## REFERÊNCIAS BIBLIOGRÁFICAS

- Abreu, L. (2011). *HTML 5*. Lisboa: FCA.
- Allied Command Transformation. (2012). *NATO C3 Classification Taxonomy*. Norfolk.
- Allied Command Transformation. (2012). *NATO C3 Classification Taxonomy*. Norfolk.
- Almeida, F. (2011). *Criar Programas* . Obtido em 15 de Novembro de 2012, de Criar Programas : <http://www.criarprogramas.com/2011/04/como-criar-sites-%E2%80%93-ferramentas-utilizadas-no-desenvolvimento-web/>
- Android. (2012). *Android*. Obtido em 12 de setembro de 2012, de Android: <http://www.android.com>
- Apple. (2012). *APPLE*. Obtido de APPLE: <http://www.apple.com/ios/>
- Barizon, B.-H. (2004). *Medidas de Propagação em 2.45GHZ para Planejamento de Redes Locais de Acesso sem fio* . Rio de Janeiro: Universidade Catolica do Rio de Janeiro.
- Barros, T. (2012). *Techtudo*. Obtido de Techtudo: <http://www.techtudo.com.br/artigos/noticia/2011/12/o-que-e-smartphone-e-para-que-serve.html>
- Boavida, F., & Bernardes, M. (2012). *TCP IP Teoria e Prática*. Lisboa: FCA.
- Bué, I. (11 de dezembro de 2012). (A. António, & M. do Carmo, Entrevistadores)
- Bué, I. (13 de Junho de 2013). Realização dos Testes ao Sistema no SIMNAV. (A. António, Entrevistador)
- Caldeira, F. (2002). *Introdução aos Sistemas de Gestão de Bases de Dados Microsoft ACCESS*. Viseu : Escola Superior de Tecnologia de Viseu .
- Coelho, P. (2004). *HTML4 & XHTML*. Lisboa: FCA.
- Coelho, P. (2005). *Programação com JSP Curso Completo*. Lisboa: FCA.
- Costa, N., Duarte , A., & Filho, N. (2011). *Avaliação Comparativa de Sistemas Operacionais para Dispositivos Móveis: Foco em Suas Funcionalidades*. Obtido em 11 de Outubro de 2012, de Tecsi: <http://www.tecsi.fea.usp.br>
- Couto, R. (2012). *Percepção e cibercepção na leitura digital : critérios a adoptar para melhorar a eficácia da leitura de artigos nos tablets*. Porto: Universidade do Porto.
- Damas, L. (2005). *Tecnologias de Informação SQL* (13ª ed.). Lisboa: FCA.
- Darcey , L., & Shane , C. (2012). *Android Wireless Application Development Advanced topics* (Vol. II ). Indiana: Addison Wesley.

- DevMedia. (2011). *DevMedia*. Obtido em 29 de novembro de 2012, de DevMedia: <http://www.devmedia.com.br/conheca-o-apache-tomcat/4546>
- Diogo, J., & Graça, R. (2011). *Java Server Pages*. Sistemas Distribuídos e Tolerância a Falhas.
- Dykes, L., & Tittel. (2005). *XML for Dummies*. Nova Jersey: Wiley.
- Fling, B. (2009). *Mobile design and development*. Bijing : O' Rielly .
- Freed, N., & Borenstein, N. (1996). *tool*. Obtido em 11 de novembro de 2012, de tool: <http://tools.ietf.org/html/rfc2045>
- Hackenhaar, J., Zanella, R., & Cardoso, T. (Dezembro de 2010). Um comparativo entre PHP e JSP: definindo a melhor aplicação para o desenvolvimento de projetos web. *Revista iTEC*, pp. 32-36.
- Heitlinger, P. (2001). *O Guia Prático da XML*. Porto: Centro Atlântico .
- IDC. (2012). *IDC*. Obtido em 12 de novembro de 2012, de IDC : <http://www.idc.pt/about/>
- IEEE. (2012). *IEEE Xplore Digital Library*. Obtido em 13 de outubro de 2012, de IEE: <http://www.ieee.org/>
- Korowajczuk, L. (2011). *WiMAX and WLAN Network Design, Optimization and Performance Analysis*. Chister : Wiley.
- Kurose, J. F., & Ross, K. W. (2007). *Computer Networking, A Top-Down Approach Featuring the Internet* (4ª ed.). Prentice Hall.
- Kushki, A., Plataniotis, K., & Venetsanopoulos, A. (2012). *WLAN Positioning Systems: Principles and Applications in Local-Based Services*. Cambrige : Cambridge University Press.
- Lee, W.-M. (2012). *Android 4 Application Development*. Indinapolis, Indiana, Estados Unidos : Wrox.
- Lima, L., Soares, L., & Endler, M. (2004). *WiMax, Padrão IEEE 802.16 para Banda Larga Sem Fios*. Rio de Janeiro: Universidade Católica do Rio de Janeiro .
- Microsoft . (2012). *Microsoft*. Obtido de Microsoft : <http://www.microsoft.com/windowsphone/en-us/default.aspx>
- Microsoft . (2012). *Office*. Obtido em 27 de novembro de 2012, de Office: <http://office.microsoft.com/pt-pt/access-help/novidades-no-microsoft-access-HA010342117.aspx#BMformandreport>
- Miranda, L. (2010). *leomiranda website & Design*. Obtido em 26 de novembro de 2012, de leomiranda website & Design: <http://www.leomiranda.com/dicas-html/mime-types-o-que-e-e-para-que-serve.html>

- Moreira, J. (2005). *WsQL, Uma Arquitectura de Software Baseada em Web Services*. Porto: Universidade do Porto.
- Patrocínio, J. (2004). *Tornar-se Pessoa e Cidadão Digital*. Almada: Universidade Nova de Lisboa.
- Pereira, A., & Poupá, C. (2011). *Linguagens WEB* (4ª ed.). Lisboa: Silabo, Lda.
- Pinheiro, C. (2011). *Dicionário do Ebook*. Sintra.
- Pinheiro, J. (2010). Obtido de Projeto de Redes: [http://www.projetoederedes.com.br/aulas/ugb\\_infraestrutura/UGB\\_aula1\\_Conceitos\\_de\\_Infraestrutura.pdf](http://www.projetoederedes.com.br/aulas/ugb_infraestrutura/UGB_aula1_Conceitos_de_Infraestrutura.pdf)
- Pinheiro, J. (Outubro de 2012). Curso de Tecnologia em Redes de Computadores. Centro Universitario Geraldo Di Biase (UGB), Rio de Janeiro, Brasil.
- Racoma. (2012). *Gmanetwork*. Obtido em novembro de 2012, de <http://www.gmanetwork.com/news/story/272698/scitech/technology/russia-makes-own-secure-android-like-tablet>
- Reis, F. L. (2010). *Como Elaborar uma Dissertação de Mestrado Segundo Bolonha*. Lisboa: Lidel.
- Ribeiro, N. (2012). *Multimédia e Tecnologias Interativas* (5ª ed.). Lisboa: FCA.
- Sanches, C. (14 de dezembro de 2012). (M. António, & G. Carmo, Entrevistadores)
- Santos, N. P. (2010). Sistema de Telemetria Aplicado a uma Plataforma Naval. *Aplicação para monitorização em tempo real dos fenómenos decorrentes do impacto de mísseis*. Alfeite, Setubal, Portugal: Escola Naval.
- Sarmiento, M. (2008). *Guia Prático sobre a Metodologia Científica para Elaboração, Escrita e Apresentação de Teses de Doutoramento, Dissertações de Mestrado e Trabalhos de Investigação Aplicada*. Lisboa, Portugal: Universidade Lusitana de Lisboa.
- Silva, J. (2012). *dspace*. Obtido em Junho de 2012, de IST: <https://dspace.ist.utl.pt/bitstream/2295/48154/1/Electronica-Basica-v1.pdf>
- Souza, A. (2011). *DevMedia*. Obtido em 29 de novembro de 2012, de DevMedia: <http://www.devmedia.com.br/servidor-web-apache-o-que-esperar/7096>
- Vasconcelos, L. E. (02 de Junho de 2013). *4learn*. Obtido em 02 de Junho de 2013, de 4learn: <http://www.4learn.pro.br/guarino/fes/Capitulo%207%20-%20Estudo%20de%20Viabilidade.pdf>
- W3C. (1999). Obtido em novembro de 2012, de W3C: <http://www.w3.org/Protocols/rfc2616/rfc2616.html>

W3C. (1996). Obtido em novembro de 2012, de W3C:  
<http://www.w3.org/Protocols/rfc1945/rfc1945>

W3C. (2005). Obtido de W3C: <http://www.w3.org/XML/>

Wi-Fi Alliance. (2012). *Wimax*. Obtido em outubro de 2012, de <http://www.wi-fi.org/>

Wimax . (2012). Obtido em outubro de 2012, de Wimax.

## ANEXOS

### Anexo A – Dispositivos móveis com as características adequadas para o projeto

Características	Smartphones			Tablets		
	<b>Samsung Galaxy S III</b> 	<b>HTC One X+</b> 	<b>Casio Gz'One Commando</b> 	<b>Samsung Galaxy Tab 2 10.1"</b> 	<b>Asus PadFone</b>  <i>Smartphone/Tablet</i>	<b>Panasonic ToughPad A1</b> 
Tipo	Normal	Normal	<i>Rugged</i>	Normal	Híbrido	<i>Rugged</i>
Peso	133 gr	135 gr	155 gr	587 gr	129/ 724 gr	950 gr
Espessura	8,6 mm	8,9 mm	15,2 mm	9,7 mm	9,2/ 13,55 mm	17,8 mm
Altura x Largura	137 x 71 mm	134 x 70 mm	130 x 66 mm	175 x 257 mm	128 x 65/ 273 x 177 mm	267 x 213
Tamanho ecrã	4,8"	4,7"	3,6"	10,1"	4,3"/ 10,1"	10,1"
Resolução ecrã	720 x 1280 pixéis	720 x 1280 pixéis	480 x 800 pixéis	1028 x 800 pixéis	540 x 960/ 800 x 1280 pixéis	1024 x 768 pixéis
Autonomia	2100 mAh	2100 mAh	1460 mAh	7000 mAh	1520/ 6600mAh	4590 mAh
Memória	16/32/64 GB	64 GB	32 GB	16/32/64 GB	32 GB	16 GB
RAM	1 GB	1 GB	512 MB	1 GB	1 GB	1 GB
CPU	Exynos 4 1,4Ghz	Nvidia Tegra 3 1,7 GHz	Qualcomm S2 800Mhz	OMAP 4330 1 GHz	Qualcomm S4 1,5 GHz	Marvell 1,2 GHz
Conectividade	Wi-Fi 801.11 a/b/g/n	Wi-Fi 801.11 a/b/g/n	Wi-Fi 801.11 b/g/n	Wi-Fi 801.11 a/b/g/n	Wi-Fi 801.11 b/g/n	Wi-Fi 801.11 a/b/g/n
SO	Android 4.0	Android 4.0	Android 4.0	Android 4.0	Android 4.0	Android 4.0

**Dados percentuais, disponibilizados pela IDC, sobre quota do mercado mundial dos principais fabricantes de *smartphones***

	Samsung	Apple	RIM	ZTE	HTC	Outros
3TRIM 2011	22,70%	13,80%	9,60%	3,30%	10,30%	40,30%
4TRIM 2011	22,50%	23,00%	8,10%	4,00%	6,40%	36,00%
1TRIM 2012	28,90%	23,00%	6,40%	4,00%	4,50%	33,20%
2TRIM 2012	32,30%	16,70%	4,90%	4,10%	5,90%	36,10%
3TRIM 2012	31,30%	15,00%	4,30%	4,20%	4,00%	41,20%

**Dados percentuais, disponibilizados pela IDC, sobre quota do mercado mundial dos principais sistemas operativos móveis**

	Android	iOS	BlackBerry OS	Symbian	Windows Phone 7/ Windows Mobile	Outros
3TRIM 2011	57,50%	13,80%	9,50%	14,60%	1,20%	3,40%
4TRIM 2011	52,80%	23,00%	8,10%	11,60%	1,20%	3,30%
1TRIM 2012	59,00%	23,00%	6,40%	6,80%	2,30%	2,50%
2TRIM 2012	69,10%	16,70%	4,90%	4,20%	3,40%	1,70%
3TRIM 2012	75,00%	14,90%	4,30%	2,30%	2,00%	1,50%

## Anexo B - SOCS utilizadas para o desenvolvimento das Aplicações

### STANDARD OPERATOR CHECKS Oficial de Quarto à Ponte

EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
<b>PCTI 11 OF/PTE</b>	<p><b>Acção Preliminar:</b> Antes de dar início ao período de SOC's, preencha a tabela existente no passo 9, utilizando a informação constante no Plano de Comunicações elaborado para a presente missão. Verifique que o PCTI se encontra desligado, i. é, com o interruptor de alimentação na posição "0" (caso esteja estabelecido, desligue-o).</p> <p><b>(1)</b> Estabeleça o PCTI, colocando o interruptor de alimentação na posição "1", e observe no display a ocorrência da seguinte sequência:  <b>(a)</b> Legenda AUTO-TESTE?  <b>(b)</b> Número designador do PCTI seguido da legenda INIC?</p> <p><b>(2)</b> Se antes de ser desligado, o PCTI <u>não estivesse</u> com comunicações estabelecidas, visualizaria:  <b>(a)</b> 11 OF/PTE?</p> <p><b>(3)</b> Se antes de ser desligado, o PCTI <u>estivesse</u> com comunicações estabelecidas:  <b>(a)</b> Restabelece automaticamente a ligação existente e indica a Conferência ou o número, seguido do nome designador do outro PCTI (INT) ou da linha em uso (RAD)?</p> <p><b>(4)</b> A lâmpada indicadora de falha está apagada?</p> <p><b>(5)</b> Premindo o botão TESTE, acendem-se todos os indicadores luminosos e todos os pontos da matriz de caracteres do display?</p> <p><b>(6)</b> Posicione o selector INT/RAD na posição INT. Utilizando os comutadores de selecção das dezenas e das unidades, o número e a legenda da estação seleccionada varia?</p> <p><b>(7)</b> Comutando o selector INT/RAD, há alteração das legendas entre circuitos internos e linhas de rádio?</p> <p><b>(8)</b> Com o selector INT/RAD comutado para a posição INT, verifique a atribuição de conferências ao PCTI e proceda ao check de comunicações internas. As comunicações estão em boas condições?</p> <p><b>(9)</b> Com o selector INT/RAD comutado para a posição RAD, verifique a atribuição de linhas rádio ao PCTI.</p>				

	<p><b>(a)</b> A atribuição das linhas no PCTI está correcta?</p> <table border="1"> <tr> <td><b>RAD 1</b></td> <td><b>RAD 2</b></td> <td><b>RAD 3</b></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table> <table border="1"> <tr> <td><b>RAD 4</b></td> <td><b>RAD 5</b></td> <td><b>RAD 6</b></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table> <table border="1"> <tr> <td><b>RAD 7</b></td> <td><b>RAD 8</b></td> <td><b>RAD 9</b></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table> <p><b>(10)</b> Selecciona a primeira linha disponível. Ouve ruído de fundo e os indicadores RAD e LIG estão acesos?</p> <p><b>(11)</b> Premindo o PTT, o indicador RAD TX acende?</p> <p><b>(12)</b> Repita as verificações 10 e 11, seleccionando as restantes linhas disponíveis.</p>	<b>RAD 1</b>	<b>RAD 2</b>	<b>RAD 3</b>				<b>RAD 4</b>	<b>RAD 5</b>	<b>RAD 6</b>				<b>RAD 7</b>	<b>RAD 8</b>	<b>RAD 9</b>							
	<b>RAD 1</b>	<b>RAD 2</b>	<b>RAD 3</b>																				
	<b>RAD 4</b>	<b>RAD 5</b>	<b>RAD 6</b>																				
	<b>RAD 7</b>	<b>RAD 8</b>	<b>RAD 9</b>																				



EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
<b>KH 1007 NTD</b>	<p>Para a total execução deste SOC, a consola a testar deve ser MASTER.</p> <p>(Se necessário premir “REQUEST MASTER”)</p> <p>(1) No Display deverá assegurar as seguintes condições de modo a obter um bom desempenho no SOC:</p> <p>(a) GANHO entre 70 e 80;</p> <p>(b) SEA no mínimo nunca zero;</p> <p>(c) RAIN no mínimo.</p> <p>(2) No menu VIDEO CORRELATION verificar se estão seleccionados os seguintes parâmetros:</p> <p>(a) INTERFERENCE REJECTION;</p> <p>(b) SCAN/SCAN CORRELATION.</p> <p>(3) Seleccionar a escala 12NM. No menu SELECT RADAR seleccionar TX MON VIDEO ON</p> <p>Aparece um “Sol” no centro do Radar cujo valor típico é 1NM e um “Pluma” com um comprimento típico de 6NM.</p> <p>(4) Verifique que o Valor do HEADING lido no display (canto superior direito) é igual ao Repetidora da Giro.</p> <p>(5) Verifique que o valor de STW (velocidade) no display (canto superior direito) é igual ao da repetidora de Odómetro.</p> <p>(6) Compare a posição GPS indicada no canto inferior direito, com a posição do respectivo GPS seleccionado.</p>				
<b>GPS PPS (TRIMBLE)</b>	<p>(1) Estabeleça o equipamento, pressionando na tecla ON.</p> <p>(a) O ecrã liga?</p> <p>(2) Após inicialização do sistema, verifique se o display LCD se encontra ligado e com indicação de valores, sendo possível a sua leitura.</p> <p>(3) Prima a tecla POS até aparecer a posição geográfica no display:</p> <p>(a) Foi possível correr o menu?</p> <p>(b) A indicação AGPS encontra-se seleccionada no canto superior esquerdo?</p> <p>(b) (c) A posição geográfica coincide com a posição indicada no display da MINS LASER?</p>				

EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
TX/RX VHF SEALAND 66	<p>(1) Estabeleça o equipamento, colocando o comutador ON/OFF na posição ON.</p> <p>(a) Aparece um número de canal no display alfanumérico?</p> <p>(b) O painel frontal é iluminado?</p> <p><b>NOTA:</b> Caso não se verifiquem as duas alíneas anteriores, verifique o ajuste do DIMMER e/ou coloque o comutador BRIDGE WINGS em LOCAL.</p> <p>(2) Ouve ruído de fundo?</p> <p><b>NOTA:</b> Caso não ouça ruído de fundo, coloque o comutador E.L.U. ON/OFF/LOUDSPEAKER ON em LOUDSPEAKER ON, rode o potenciômetro de SQUELCH para a direita e ajuste o volume (potenciômetro VOLUME).</p> <p>(3) Rode o potenciômetro SQUELCH para a esquerda até desaparecer o ruído de fundo.</p> <p>(4) Selecione um canal, premindo a tecla CLEAR e o número do canal.</p> <p>Apareceu no display o número do canal seleccionado?</p> <p>(5) Selecione a potência desejada (POT LOW/HIGH) e prima o PTT do Handset.</p> <p>Acende-se a lâmpada indicadora de POT HIGH?</p> <p>(6) Verifique se o equipamento executa DUAL WATCH, procedendo da seguinte forma:</p> <p>Selecione dois canais para DUAL WATCH, premindo as teclas A e CLEAR, seguido do número do canal.</p> <p>Seguidamente prima as teclas B e CLEAR, seguido do número do canal.</p> <p>Prima a tecla DUAL WATCH.</p> <p>(a) No Display aparecem dois pontos a piscar alternadamente?</p> <p>(b) Logo que for recebido um sinal em qualquer destes canais, o display assinala o canal em uso?</p>				

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32509	VHF SAILOR	<p>(1) Estabeleça o equipamento, rodando o potenciômetro de VOLUME no sentido dos ponteiros do relógio.</p> <p>(a) Aparece um número de canal no display alfanumérico?</p> <p>(b) O painel frontal é iluminado?</p> <p><b>NOTA:</b> Caso não se verifiquem as duas alíneas anteriores, verifique o ajuste do DIMMER, pressionando SHIFT e "7" (DIM).</p> <p>(2) Ouve ruído de fundo?</p> <p><b>NOTA:</b> Caso não ouça ruído de fundo, rode o potenciômetro de SQUELCH para a esquerda e ajuste o volume (potenciômetro VOLUME).</p> <p>(3) Rode o potenciômetro SQUELCH para a direita até desaparecer o ruído de fundo.</p> <p>(4) Selecione um canal, premindo a(s) tecla(s) do número do canal. Apareceu no display o número do canal introduzido?</p> <p>(5) Premindo SHIFT e "9" (1W), acende-se a lâmpada indicadora de potência "1W"?</p> <p>(6) Selecione a potência desejada (1W/5W) premindo SHIFT e "9" (1W).</p> <p><b>NOTA:</b> O equipamento transmite com 5W quando a lâmpada "1W" está apagada.</p> <p>(7) Prima o PTT do Handset. Acende-se a lâmpada indicadora de TX?</p> <p>(8) Verifique se o equipamento executa DUAL WATCH, procedendo da seguinte forma:</p> <p>(a) Selecione o segundo canal para DUAL WATCH, premindo a(s) tecla(s) do número do canal pretendido.</p> <p>(b) Prima a tecla SHIFT e "6" (DW).</p> <p>(c) No display aparecem "16" e o número do segundo canal a piscar alternadamente?</p> <p>(d) Logo que for recebido um sinal em qualquer destes canais, o display assinala o canal em uso?</p>				

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
XXXXX	DSC VHF SAILOR	<p><b>Nota:</b> Para se realizar esta Soc a impressora do DSC tem que estar ligada.</p> <p>(1) Estabeleça o equipamento, rodando o potenciômetro para a direita.</p> <p>(a) Aparece no ecrã WATCHING CH 70 e a hora ?</p> <p>(b) Confirme a hora com a informação dada pelo GPS</p> <p>(2) Pressione em FUNC. Pressione em "&gt;" até a palavra TEST ficar a piscar. Pressione em NEXT.</p> <p>(a) O equipamento passou no teste? Pressione em NEXT</p> <p>3. Pressione em CALL. Selecione FRACABRAL. Pressione em NEXT duas vezes. Pressione em SEND</p> <p>(a) O equipamento começa a apitar e dar informação de mensagem recebida?</p> <p>(b) A impressora imprime a informação da mensagem?</p> <p>4. Carregue em NEXT. Carregue em LAST duas vezes.</p>				

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32281	ECPINS-W 5.1	<p><b>(1)</b> Seleccionar <b>main menu »Screen Layout» alarms layout</b>. As indicações (canto inferior esquerdo) de “Lat/Lon”, “HDG”, SMG/CMG”, “Slog”, “Depth”, apresentam valores correctos?</p> <p><b>(a)</b> Indicação,... comparar com,..... diferença.  “HDG” Repetidora da giro _____  (lme)  “SLog” Odómetro _____  “CMG” “COG” no GPS/DGPS _____  “SMG” “SOG” no GPS/DGPS _____  “Lat/Lon” GPS/DGPS _____  “Depth” Sonda _____</p> <p><b>(2)</b> Existem mensagens de erro no Painel de Alarms (vermelho/amarelo)?</p> <p><b>(3)</b> Ao seleccionar uma “<b>Route</b>” (preferencialmente a que será usada na saída do porto), verificar se aparecem as informações de:  “Nome da Route”/”wp 1 to wp 2”  “XTD” / “DTG”  “WP BRG” / “TTG”  “Leg Course”  “Dest ETA”  “Next CTS”</p> <p><b>(4)</b> Ao variara posição do cursor na carta também variam os seguintes dados (no painel <b>CURSOR</b>)?  Coordenadas do cursor  “RNG”  “BNG”  “Mag Var”  “TTG”</p> <p><b>(5)</b> Variar o brilho e contraste do monitor seleccionado, <b>main menu» color scheme</b>  <b>(a)</b> Seleccionar <b>main menu» setup» display setup» Brightness Contrast</b>  Actuar nos comandos de modo a obter uma imagem adequada à luminosidade no interior da ponte.  <b>(b)</b> Os indicadores apresentam fundo preto e dígitos a amarelo?</p>				

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32457	MINS LASER	(1) Estabeleça o equipamento, pressionando na tecla ON. (a) O ecrã liga? (2) Pressione a tecla “2”, MINS STATE. (a) As condições CDU e ICU encontram-se em ON? (b) O modo DRU está em NAVIGATION? (3) Pressione a tecla “3” (a) Se o GPS “Trimble” estiver disponível compare os valores INT HEADING, SPEED e LAT, LON. são coincidentes? (b) Se o GPS não estiver disponível esses valores são coerentes com a actual situação e posição do navio?				
XXXXX	NAVTEX	(1) Estabeleça o equipamento, colocando o comutador ON/OFF na posição ON. (a) A luz indicadora SIGNAL 518 está acesa? (2) Abra a tampa colocada à frente do equipamento e coloque a patilha TEST/SETUP em TEST. (1) O teste dá a indicação PASS em todos os passos? (3) Feche a tampa.				
32508	RADIOGONIÓMETRO VHF	(1) Ligue o equipamento. Há alguma indicação nos displays? (Se negativo, actue no comando DIMMER) (2) Actue no comando VOLUME. (a) Existe alguma variação no ruído de fundo? (3) Actue no comando SQUELCH. (a) Existe alguma variação no ruído de fundo? (4) Prima o comando MANU. Prima a tecla F e introduza os dígitos 1, 5, 6, 7, 0 e 0. (a) Os números são mostrados no indicador de frequência? (5) Prima a tecla E.				

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
	DGPS FURUNO	<p>(1) Estabeleça o equipamento, pressionando o botão POWER. (a) O ecrã liga?</p> <p>(2) Premir tecla <b>DISPLAY</b>, seleccionar <b>DATA</b>, premir <b>ENTER</b>. Aparece info. de pst geográfica?</p> <p>(3) A indicação <b>DGPS 3D</b> aparece no canto superior direito? (só aplicável em áreas com cobertura DGPS)</p> <p>(4) Premir tecla <b>MENU   ESC</b>, aceder opção de menu <b>8. Self Tests</b>, aceder opção <b>4. Automatic Testing</b>. Premir <b>ENT</b>. Todos os resultados OK?</p>				

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32507	RADIOGONIÓMETRO HF/MF	<p>(1) Verifique se o equipamento está ligado. Caso esteja, desligue-o.</p> <p>(2) Coloque o comando SYSTEM em RCV e verifique:            (a) O indicador de CHANNEL mostra “00”?            (b) O indicador de frequência mostra “2182.0”?            (Caso negativo, actue no comando DIMMER)</p> <p>(3) Actue no comando VOLUME.            (a) Existe alguma variação no ruído de fundo?</p> <p>(4) Actue no comando RF GAIN.            (a) Existe variação de amplitude na imagem do T. R. C:?</p> <p>(5) Prima o comando MANU.</p> <p>(6) Prima a tecla F e introduza os dígitos 1, 2, 3, 4, 5 e 6.            (a) Os números são mostrados no indicador de frequência?</p> <p>(7) Prima a tecla E.</p> <p>(8) Gire o botão TUNE            (a) A frequência introduzida nos passos 6 e 7 varia?</p> <p>(9) Actue as teclas ↑ e ↓.            (a) A frequência varia?</p>				
32270	AIS	<p>(1) Estabeleça o equipamento, pressionando na tecla POWER.            (a) O ecrã liga?</p> <p>(2) Após inicialização do sistema, verifique se o display LCD se encontra ligado e com indicação de valores, sendo possível a sua leitura.</p> <p>(3) Aparece no canto superior esquerdo a posição geográfica?</p> <p>(4) Aparece no lado esquerdo do LCD informação reativa a contactos: MMSI, Nome, Azimute, Distância, Velocidade e Rumo?</p> <p>(5) Prima DISPLAY há alteração no LCD?</p>				

## SRBOC

NOTA: Nos dias pares será testado o lançador de BB.  
Nos dias impares, o lançador de EB.

Procedimento:

(1) Aguarde até o CO indicar para dar início ao SOC do SRBOC a partir da Ponte.

(2) Teste comando a partir da Ponte.

(3) Abra a tampa metálica da unidade de comando do lançador em teste e verifique que as lâmpadas SHIPS POWER e BRIDGE estão acesas.

(4) Se a unidade de comando não estiver estabelecida, prima POWER OFF e verifique o seguinte:

(a) A lâmpada POWER ON fica acesa?

(b) O indicador de Arm indica SAFE?

(c) O indicador de Control indica BRIDGE;?

(d) Todos os indicadores de status dos tubos indicam EMPTY?

(5) Informe o municiador SRBOC que está pronto para iniciar os testes ao SRBOC BB/EB.

(6) Ordene a colocação da cartridge de teste num dos tubos (1, 2, 3, 4, 5 e 6).

NOTA: A rotatividade dos tubos deve ser assegurada para que o teste seja efectuado a todos os tubos.

(7) Quando todas as condições estiverem reunidas efectue tentativa de fogo no tubo 1, execute/verifique o seguinte:

(a) A lâmpada de status do tubo 1 mudou de EMPTY para LOADED;

(b) O indicador Arm indica SAFE;

(c) Prima e mantenha o switch de ARM; assegure-se que a lâmpada ARM ON acende;

(d) Prima e largue o swith do tubo em causa (LOADED);

(e) Largue o switch de ARM e verifique que volta a indicar SAFE;

(f) Após o municiador retirar do tubo a Cartridge de teste, verifique que a lâmpada de status do tubo passou para EMPTY.

(8) Repita os passos (a) a (f) da alínea 7 para os tubos 2 a 4.

Nos tubos 5 e 6 (IR) repita os passos (a) a (f) com a alteração do passo (c), em que deve manter o switch de ARM premido e esperar que o indicador de teste Cartridge indique 5 disparos (pode demorar entre 10 a 100 segundos).



**GDH SOC'S**
**REALIZADOS:** \_\_\_\_\_ **OQP:** \_\_\_\_\_

**STANDARD OPERATOR CHECKS Sargento de Quarto à Ponte**

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32556	PCTI 09 CMD/PTE	<p><b>Ação Preliminar:</b> Antes de dar início ao período de SOC's, preencha a tabela existente no passo 9, utilizando a informação constante no Plano de Comunicações elaborado para a presente missão. Verifique que o PCTI se encontra desligado, i. é, com o interruptor de alimentação na posição "0" (caso esteja estabelecido, desligue-o).</p> <p>(1) Estabeleça o PCTI, colocando o interruptor de alimentação na posição "1", e observe no display a ocorrência da seguinte sequência:  (a) Legenda AUTO-TESTE?  (b) Número designador do PCTI seguido da legenda INIC?</p> <p>(2) Se antes de ser desligado, o PCTI <u>não estivesse</u> com comunicações estabelecidas, visualizaria:  (a) 09 CMD/PTE?</p> <p>(3) Se antes de ser desligado, o PCTI <u>estivesse</u> com comunicações estabelecidas:  (a) Restabelece automaticamente a ligação existente e indica a Conferência ou o número, seguido do nome designador do outro PCTI (INT) ou da linha em uso (RAD)?</p> <p>(4) A lâmpada indicadora de falha está apagada?</p> <p>(5) Premindo o botão TESTE, acendem-se todos os indicadores luminosos e todos os pontos da matriz de caracteres do display?</p> <p>(6) Posicione o selector INT/RAD na posição INT. Utilizando os comutadores de selecção das dezenas e das unidades, o número e a legenda da estação seleccionada varia?</p> <p>(7) Comutando o selector INT/RAD, há alteração das legendas entre circuitos internos e linhas de rádio?</p> <p>(8) Com o selector INT/RAD comutado para a posição INT, verifique a atribuição de conferências ao PCTI e proceda ao check de comunicações internas. As comunicações estão em boas condições?</p> <p>(9) Com o selector INT/RAD comutado para a posição RAD, verifique a atribuição de linhas rádio ao PCTI.  (a) A atribuição das linhas no PCTI está correcta?</p>				

Nº	EQUIPAMENTO	SOC			OP	DEG	INOP	OBS
32556		RAD 1	RAD 2	RAD 3				
		RAD 4	RAD 5	RAD 6				
		RAD 7	RAD 8	RAD 9				
		<p>(10) Seleccione a primeira linha disponível. Ouve ruído de fundo e os indicadores RAD e LIG estão acesos?</p> <p>(11) Premindo o PTT, o indicador RAD TX acende?</p> <p>(12) Repita as verificações 10 e 11, seleccionando as restantes linhas disponíveis.</p>						

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32587	PCTI 10 CMD/PTE	<p><b>Acção Preliminar:</b> Antes de dar início ao período de SOC's, preencha a tabela existente no passo 9, utilizando a informação constante no Plano de Comunicações elaborado para a presente missão.</p> <p>Verifique que o PCTI se encontra desligado, i. é, com o interruptor de alimentação na posição "0" (caso esteja estabelecido, desligue-o).</p> <p>(1) Estabeleça o PCTI, colocando o interruptor de alimentação na posição "1", e observe no display a ocorrência da seguinte sequência:  (a) Legenda AUTO-TESTE?  (b) Número designador do PCTI seguido da legenda INIC?</p> <p>(2) Se antes de ser desligado, o PCTI <u>não estivesse</u> com comunicações estabelecidas, visualizaria:  (a) 10 CTG/PTE?</p> <p>(3) Se antes de ser desligado, o PCTI <u>estivesse</u> com comunicações estabelecidas:  (a) Restabelece automaticamente a ligação existente e indica a Conferência ou o número, seguido do nome designador do outro PCTI (INT) ou da linha em uso (RAD)?</p> <p>(4) A lâmpada indicadora de falha está apagada?</p> <p>(5) Premindo o botão TESTE, acendem-se todos os indicadores luminosos e todos os pontos da matriz de caracteres do display?</p> <p>(6) Posicione o selector INT/RAD na posição INT. Utilizando os comutadores de selecção das dezenas e das unidades, o número e a legenda da estação seleccionada varia?</p> <p>(7) Comutando o selector INT/RAD, há alteração das legendas entre circuitos internos e linhas de rádio?</p> <p>(8) Com o selector INT/RAD comutado para a posição INT, verifique a atribuição de conferências ao PCTI e proceda ao check de comunicações internas.  As comunicações estão em boas condições?</p> <p>(9) Com o selector INT/RAD comutado para a posição RAD, verifique a atribuição de linhas rádio ao PCTI.  (a) A atribuição das linhas no PCTI está correcta?</p>				
Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS

32587		<table border="1"> <tr> <td>RAD 1</td> <td>RAD 2</td> <td>RAD 3</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>	RAD 1	RAD 2	RAD 3							
		RAD 1	RAD 2	RAD 3								
		<table border="1"> <tr> <td>RAD 4</td> <td>RAD 5</td> <td>RAD 6</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>	RAD 4	RAD 5	RAD 6							
		RAD 4	RAD 5	RAD 6								
		<table border="1"> <tr> <td>RAD 7</td> <td>RAD 8</td> <td>RAD 9</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>	RAD 7	RAD 8	RAD 9							
		RAD 7	RAD 8	RAD 9								
		<p>(10) Seleccione a primeira linha disponível. Ouve ruído de fundo e os indicadores RAD e LIG estão acesos?</p> <p>(11) Premindo o PTT, o indicador RAD TX acende?</p> <p>(12) Repita as verificações 10 e 11, seleccionando as restantes linhas disponíveis.</p>										

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32590	PCTI 40 SAR/PTE	<p>Verifique se o PCTI se encontra desligado, i. é, com o interruptor de alimentação na posição "0" (caso esteja estabelecido, desligue-o).</p> <p><b>(1)</b> Estabeleça o PCTI colocando o interruptor de alimentação na posição "1" e observe no display a ocorrência da seguinte sequência:</p> <p><b>(a)</b> Legenda AUTO-TESTE?</p> <p><b>(b)</b> Número designador do PCTI seguido da legenda INIC?</p> <p><b>(2)</b> Se antes de ser desligado, o PCTI <u>não estivesse</u> com comunicações estabelecidas, visualizaria:</p> <p><b>(a)</b> 40 SAR/PTE?</p> <p><b>(3)</b> Se antes de ser desligado, o PCTI <u>estivesse</u> com comunicações estabelecidas:</p> <p><b>(a)</b> Restabelece automaticamente a ligação existente e indica a Conferência ou o número, seguido do nome designador do outro PCTI?</p> <p><b>(4)</b> A lâmpada indicadora de falha está apagada?</p> <p><b>(5)</b> Premindo o botão TESTE, acendem-se todos os indicadores luminosos e todos os pontos da matriz de caracteres do display?</p> <p><b>(6)</b> Usando os comutadores de selecção das dezenas e das unidades, o número e a legenda da estação seleccionada varia?</p> <p><b>(7)</b> Verifique a atribuição de conferências ao PCTI e proceda ao check de comunicações internas. As comunicações estão em boas condições?</p>				

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32462	GIRO	<p>No quadro NAUTOALARM localizado na consola por cima da mesa das cartas, proceda às seguintes verificações:</p> <p>(1) Ao premir a tecla TEST RESET, todos os indicadores acendem e o sinal sonoro de alarme soa?</p> <p>(2) Premindo a tecla DIM durante alguns segundos, os indicadores luminosos variam de brilho?</p> <p>(3) O quadro apresenta indicações luminosas?</p> <p>(4) As indicações presentes são todas de cor verdes?</p> <p>Proceda à comparação das diversas repetidoras da girobússola, posicionando previamente os seguintes indivíduos junto das repetidoras indicadas:</p> <p>Oficial de Quarto à Ponte: Repetidora central da Ponte</p> <p>Sargento de Quarto à Ponte: Repetidora da Consola de Navegação</p> <p>Cabo de Quarto: Repetidora da Casa da Máquina do Leme</p> <p>Vigias: Repetidora</p> <p>s das Asas da Ponte</p> <p>Elemento do DPE: Repetidora do MCR</p> <p>Através da Conferência Interna, o Marinheiro do Leme dá a voz de «FORA», simultaneamente para a Ponte, MCR e Casa da Máquina do Leme. Os diversos elementos reportam se a indicação de Proa fornecida pela Marinheiro de Leme coincide com a leitura registada e os eventuais erros verificados:</p> <p>Repetidora central da Ponte?</p> <p>Repetidora da Consola de Navegação?</p> <p>Repetidora da Casa da Máquina do Leme?</p> <p>Repetidoras das Asas da Ponte?</p> <p>Repetidora do MCR?</p> <p>(5) Rodando o potenciômetro de regulação de intensidade de brilho em todas as repetidoras da zona da Ponte, a iluminação varia de intensidade?</p>				

XXXXX

SONDA

- NOTA:** Este SOC só tem completa aplicação com o navio atracado, fundeado ou a navegar em águas pouco profundas (< 500 m).
- (1) Com o equipamento desligado e o MAIN SWITCH na posição OFF. A lâmpada HEATER encontra-se acesa?
- (2) Estabeleça o equipamento, colocando o comutador MAIN SWITCH na posição ON. A lâmpada MAINS acendeu?
- (3) Na repetidora instalada na consola do leme, coloque os reguladores de volume e de intensidade de brilho no máximo (rodar totalmente no sentido dos ponteiros do relógio). Verifique que se encontra estabelecida e não apresenta indicação de alarme.
- (4) Colocando os comutadores DISPLAY na posição DIGITAL e PULSE SEQ na posição CONT, verifique se tem indicação de profundidade nos indicadores digitais WATER DEPTH existentes na Unidade Principal (UP) e na Repetidora, e que o valor apresentado corresponde ao local onde o navio se encontra.
- (5) Proceda ao teste do alarme de profundidade:
- (a) Coloque o comutador "< >" na posição ">".  
Com o comutador "↑↓", introduza um valor menor que a profundidade indicada no display.
- i. Na UP é activado o alarme sonoro e acende o indicador MARK?
- ii. Na Repetidora é activado o alarme sonoro e acende o indicador ALARM?
- iii. Verifique que actuando nos potenciómetros de regulação de volume (UP e Rep), a intensidade do sinal sonoro varia.
- (b) Coloque o comutador "< >" na posição "<".  
Os sinais de alarme sonoros e luminosos desaparecem?
- (c) Com o comutador "↑↓", introduza um valor maior que a profundidade indicada no display. Os alarmes sonoros e luminosos são activados?
- (d) Coloque o comutador "< >" na posição ">".  
Os sinais de alarme sonoros e luminosos desaparecem?

		<p><b>(7)</b> Colocando o comutador DISPLAY na posição RECORDER+DIGITAL:</p> <p><b>(a)</b> Acende-se a iluminação do Registrador e este inicia o registo?</p> <p><b>(b)</b> Os display's WATER DEPTH ( UP e Rep) indicam leitura de profundidade que confere com a indicação no registo em papel?</p> <p><b>(8)</b> Comutando a escala RANGE entre 25 m e 500 m, a velocidade da cabeça de escrita no Registrador varia, sendo mais lenta nas posições 250 m e 500 m?</p> <p><b>(9)</b> Actue o potenciômetro DIMMER. A intensidade da iluminação no Registrador varia?</p> <p><b>(10)</b> Coloque o selector DYNAMIC na posição 6. O registo torna-se mais escuro?</p> <p><b>(11)</b> Coloque RANGE no valor de escala mínimo para a profundidade local. Coloque PULSE SEQ em SINGLE. Leve momentaneamente PULSE SEQ da posição SINGLE a "•" (tempo suficiente para o indicador vermelho acender):</p> <p><b>(a)</b> Verifique que o display na UP não indica qualquer valor;</p> <p><b>(b)</b> Verifique que o display na Repetidora apresenta um ponto decimal;</p> <p><b>(c)</b> No papel é efectuado um registo constante de duração limitada (<math>\approx</math> 25 segundos, na escala dos 25 m)?</p> <p><b>(d)</b> Verifique que o aspecto do registo se altera em cada nova transmissão, actuando no comutador PULSE SEQ levando-o da posição SINGLE a "•".</p> <p><b>(12)</b> Colocando o comutador DISPLAY na posição RECORDER:</p> <p><b>(a)</b> Os indicadores digitais de profundidade mantêm-se apagados (UP) e com ponto decimal (Rep)?</p> <p><b>(b)</b> O Registrador da Sonda continua em movimento?</p> <p><b>(13)</b> Actuando no selector APER SPEED, a velocidade do papel é maior na posição II que na posição I, e é proporcional à velocidade do navio na posição LOG?</p> <p><b>(14)</b> Premindo o botão MARK durante algum tempo, o registador marca uma linha vertical no papel do registo?</p>				
--	--	---	--	--	--	--



Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32433	ÓDOMETRO (M. CARTAS)	<p>Na unidade principal sobre a mesa das cartas, proceda às seguintes verificações:</p> <p>(1) O display indicador de Knots e Sea Miles <u>não</u> está a piscar?</p> <p>(2) Se o indicador referido em 1 estiver a piscar: Desligue o Odómetro premindo a tecla POWER. Volte a estabelecê-lo premindo novamente a tecla POWER. O display deixou de piscar?</p> <p>(3) Verifique se os seguintes indicadores se apresentam acesos:</p> <p>(a) PROBE</p> <p>(b) CURVE</p> <p>(c) MODE</p> <p>(d) P1</p> <p>(e) C1 (ou C2)</p>				
32435	ÓDOMETRO (C. TELEGRAFOS)	<p>Na repetidora instalada na consola dos Telégrafos, proceda às seguintes verificações:</p> <p>(1) Premindo simultaneamente as teclas BACKLIGHT (+ e -), todos os indicadores ficam visíveis durante alguns segundos?</p> <p>(2) Premindo as teclas BACKLIGHT (+) e BACKLIGHT (-) alternadamente, a iluminação do display varia de intensidade?</p> <p>(3) A velocidade e o número de milhas percorridas indicado no display coincidem com a indicação da unidade principal (existente na ponte)?</p>				

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32424	CCTV EB	<p>(1) Verifique se o monitor se encontra estabelecido, i. é, o botão ON premido.</p> <p>(2) Seleccione um canal diferente de 00.</p> <p>(3) O monitor tem imagem?</p> <p>(4) Os seguintes comandos actuam:</p> <p style="padding-left: 40px;">(a) BRILHO?</p> <p style="padding-left: 40px;">(b) CONTRASTE?</p> <p>(5) O monitor apresenta a lâmpada vermelha na parte inferior esquerda do painel acesa?</p> <p>Caso a resposta anterior seja NÃO, mande chamar o técnico responsável para que seja estabelecida a alimentação do monitor. Este SOC não pode prosseguir sem ter passado o ponto 5 com sucesso.</p> <p>(6) Actuando as teclas do selector de canais, verifique se os seguintes canais são seleccionáveis:</p> <p style="padding-left: 40px;">(a) CÂMARA I</p> <p style="padding-left: 40px;">(b) CÂMARA II</p> <p style="padding-left: 40px;">(c) CÂMARA III</p> <p style="padding-left: 40px;">(d) CÂMARA IV</p> <p style="padding-left: 40px;">(e) STIR AV</p> <p style="padding-left: 40px;">(f) STIR AR</p> <p style="padding-left: 40px;">(g) SIST. GRAVAÇÃO VÍDEO</p> <p style="padding-left: 40px;">(h) CCC I</p> <p style="padding-left: 40px;">(i) CCC II</p> <p style="padding-left: 40px;">(j) P.HEL I</p> <p>(7) No painel de selecção de canais, aparece a indicação correcta do canal seleccionado?</p>				

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32323	CCTV BB	<p>(1) Verifique se o monitor se encontra estabelecido, i. é, o botão ON premido.</p> <p>(2) Seleccione um canal diferente de 00.</p> <p>(3) O monitor tem imagem?</p> <p>(4) Os seguintes comandos actuam:</p> <p style="padding-left: 40px;">(a) BRILHO?</p> <p style="padding-left: 40px;">(b) CONTRASTE?</p> <p>(5) O monitor apresenta a lâmpada vermelha na parte inferior esquerda do painel acesa?</p> <p>Caso a resposta anterior seja NÃO, mande chamar o técnico responsável para que seja estabelecida a alimentação do monitor. Este SOC não pode prosseguir sem ter passado o ponto 5 com sucesso.</p> <p>(6) Actuando as teclas do selector de canais, verifique se os seguintes canais são seleccionáveis:</p> <p style="padding-left: 40px;">(a) CÂMARA I</p> <p style="padding-left: 40px;">(b) CÂMARA II</p> <p style="padding-left: 40px;">(c) CÂMARA III</p> <p style="padding-left: 40px;">(d) CÂMARA IV</p> <p style="padding-left: 40px;">(e) STIR AV</p> <p style="padding-left: 40px;">(f) STIR AR</p> <p style="padding-left: 40px;">(g) SIST. GRAVAÇÃO VÍDEO</p> <p style="padding-left: 40px;">(h) CCC I</p> <p style="padding-left: 40px;">(i) CCC II</p> <p style="padding-left: 40px;">(j) P.HEL I</p> <p>(7) No painel de selecção de canais, aparece a indicação correcta do canal seleccionado?</p>				

32447	SHIPS TIME SYSTEMS	<p>Na unidade principal localizada por cima da mesa das cartas, verifique se:</p> <p>(1) O display está aceso, apresentando indicação correcta da hora, e os dígitos dos segundos estão a contar?</p> <p>(2) A indicação no display é contínua, i. é, não pisca?</p> <p>(3) O display não apresenta nenhuma luz vermelha intermitente do lado esquerdo dos dígitos?</p> <p>(4) Os segundos apresentam uma marcha regular?</p> <p>Na repetidora colocada a vante da ponte, verifique se:</p> <p>(5) A marcha dos ponteiros é regular e coincide com a indicação fornecida no display da unidade principal?</p> <p>(6) Rodando o potenciómetro de regulação do brilho, a intensidade da iluminação varia?</p>				
-------	-----------------------	---	--	--	--	--

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32443	<b>ANEMÓMETRO (M. DAS CARTAS)</b>	<p>Nos indicadores existentes na consola por cima da mesa das cartas, proceda à seguinte verificação:</p> <p><b>(1)</b> Os indicadores de direcção e velocidade do vento apresentam variação?</p> <p><b>(2)</b> Com velocidades de vento inferiores a 25 nós, premindo o botão branco, o ponteiro deflecte para a direita, apresentando uma indicação correcta na escala interior?</p> <p><b>(3)</b> A direcção e velocidade do vento registada são idênticas à indicada nos indicadores existentes a vante da Ponte?</p> <p><b>(4)</b> Actuando o reóstato de regulação de brilho, a iluminação dos indicadores varia?</p>				
32442	<b>ANEMÓMETRO (AV DA PONTE)</b>	<p>Nos indicadores existentes a vante da Ponte, proceda à seguinte verificação:</p> <p><b>(1)</b> Os indicadores de direcção e velocidade do vento apresentam variação?</p> <p><b>(2)</b> Com velocidades de vento inferiores a 25 nós, premindo o botão branco, o ponteiro deflecte para a direita, apresentando uma indicação correcta na escala interior?</p> <p><b>(3)</b> A direcção e velocidade do vento registada são idênticas à indicada nos indicadores existentes na consola por cima da mesa das cartas?</p> <p><b>(4)</b> Actuando o reóstato de regulação de brilho, a iluminação dos indicadores varia?</p>				
21200	<b>SEREIA</b>	<p>No painel de comando, verifique o seguinte:</p> <p><b>(1)</b> O potenciómetro de controlo de intensidade luminosa actua?</p> <p><b>(2)</b> Pressione o botão superior esquerdo durante 1 segundo. O sinal sonoro actua?</p> <p><b>(3)</b> Pressione os botões existentes nas asas da Ponte e junto do VHF SEALAND. O sinal sonoro actua?</p>				

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32372	LUZES NAVEGAÇÃO	<p>(1) Colocar o selector FEEDING na posição I. Os 3 Led's verdes ON, Alim. I e Alim. II acendem?</p> <p>(2) Colocar o selector FEEDING na posição II. Os 3 Led's verdes ON, Alim. I e Alim. II acendem?</p> <p>(3) Com o selector FEEDING na posição I ou II:</p> <p>(a) Os indicadores Led's e o besouro actuam ao premir o botão REST LED?</p> <p>(b) Todos os Led's de alimentação (vermelhos) e de posição acendem ao premir o botão correspondente?</p> <p>(4) Ao rodar o reóstato DIMMER, o brilho dos Led's aumenta e diminui?</p> <p>(5) Ao comutar o selector de brilho, a intensidade das lâmpadas dos faróis/luzes de sinalização varia?</p>				

GDH

SOC'S

REALIZADOS: \_\_\_\_\_ OPERADOR: \_\_\_\_\_

## STANDARD OPERATOR CHECKS Marinheiro do Leme

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32581	PCTI 37 LEME	<p>Verifique se o PCTI se encontra desligado, i. é, com o interruptor de alimentação na posição "0" (caso esteja estabelecido, desligue-o).</p> <p><b>(1)</b> Estabeleça o PCTI colocando o interruptor de alimentação na posição "1" e observe no display a ocorrência da seguinte sequência:</p> <p style="padding-left: 40px;"><b>(a)</b> Legenda AUTO-TESTE?</p> <p style="padding-left: 40px;"><b>(b)</b> Número designador do PCTI seguido da legenda INIC?</p> <p><b>(2)</b> Se antes de ser desligado, o PCTI <u>não estivesse</u> com comunicações estabelecidas, visualizaria:</p> <p style="padding-left: 40px;"><b>(a)</b> 37 LEME?</p> <p><b>(3)</b> Se antes de ser desligado, o PCTI <u>estivesse</u> com comunicações estabelecidas:</p> <p style="padding-left: 40px;"><b>(a)</b> Restabelece automaticamente a ligação existente e indica a Conferência ou o número, seguido do nome designador do outro PCTI?</p> <p><b>(4)</b> A lâmpada indicadora de falha está apagada?</p> <p><b>(5)</b> Premindo o botão TESTE, acendem-se todos os indicadores luminosos e todos os pontos da matriz de caracteres do display?</p> <p><b>(6)</b> Usando os comutadores de selecção das dezenas e das unidades, o número e a legenda da estação seleccionada varia?</p> <p><b>(7)</b> Verifique a atribuição de conferências ao PCTI e proceda ao check de comunicações internas. As comunicações estão em boas condições?</p>				

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
32495	PILOTO AUTOMÁTICO	<p>- Sem ter o Piloto Automático seleccionado (i. é, com o governo do navio a ser feito pela roda do leme ou pelo manipulo) proceda às seguintes verificações:</p> <p>(1) Existe iluminação no display?</p> <p>(2) Premindo a tecla TEST RESET do painel superior NAUTOPILOT D, todos os indicadores iluminam-se?</p> <p>(3) Rodando o botão rotativo central, o brilho do display varia?</p> <p>(4) Premindo as teclas de cursor (seta apontada para cima e seta apontada para baixo), os indicadores que estão por cima destas teclas iluminam-se sequencialmente a cada toque, e no display com a legenda SET COURSE, aparece um valor dentro dos limites indicados para cada indicador?</p> <p>(5) Premindo a tecla TEST RESET do painel inferior NAUTOALARM, verifique se:</p> <p>(a) Todos os indicadores acendem.</p> <p>(b) O alarme sonoro soa.</p> <p>(6) A proa indicada no display HEADING coincide com a proa do navio lida nas repetidoras da giro?</p> <p>(7) Ao comutar o seleccionador AUTO PILOT/RODA DE LEME/MANÍPULO no painel inferior, o indicador respectivo acende-se quando é seleccionada a posição de:</p> <p>(a) AUTO PILOT</p> <p>(b) RODA DE LEME</p> <p>(c) MANIPULO</p>				
XXXXX	AGULHA MAGNÉTICA	<p>Na agulha magnética, verifique o seguinte:</p> <p>(1) A leitura de proa coincide com os valores obtidos em leituras anteriores para a mesma proa, na mesma área?</p> <p>(2) Caso não tenha procedido à leitura anterior, solicite ao Oficial de Quarto a indicação da declinação magnética da área, e o desvio para a proa a que se navega. A leitura obtida na agulha magnética é idêntica ao valor esperado?</p> <p>(3) A lâmpada de iluminação está acesa e, rodando o potenciômetro de controlo de intensidade, o brilho varia?</p>				



Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
XXXXX	INDICADOR ÂNGULO LEME	<p><b>Acções Preliminares:</b></p> <ul style="list-style-type: none"> <li>- Preposicionar os Vigias nas asas da Ponte e o Cabo de Quarto na Casa da Máquina do Leme.</li> <li>- Estabeleça comunicações através da conferência apropriada com a Casa da Máquina do Leme, MCR e Centro de Operações.</li> <li>- Peça autorização ao Oficial de Quarto à Ponte para verificar os indicadores de ângulo de leme.</li> </ul> <p><b>Verificação a efectuar:</b></p> <p><b>(1)</b> Introduza os seguintes ângulos de leme e informe as diferentes estações:</p> <ul style="list-style-type: none"> <li><b>(a)</b> Leme a meio;</li> <li><b>(b)</b> 10º de leme a EB;</li> <li><b>(c)</b> 10º de leme a BB.</li> </ul> <p><b>(2)</b> Verifique que nos diferentes indicadores, as leituras são correctas:</p> <ul style="list-style-type: none"> <li><b>(a)</b> Marinheiro do Leme;</li> <li><b>(b)</b> Asa da Ponte de EB;</li> <li><b>(c)</b> Asa da Ponte de BB;</li> <li><b>(d)</b> CO;</li> <li><b>(e)</b> MCR;</li> <li><b>(f)</b> Indicador da Máquina do Leme;</li> <li><b>(g)</b> Axiómetro do Aparelho do Leme.</li> </ul>				

Nº	EQUIPAMENTO	SOC	OP	DEG	INOP	OBS
31133 / 31144	SBROC (MUNICIADOR)	<p>NOTA: Nos dias pares será testado o lançador de EB (CO) e BB (Ponte). Nos dias impares, o lançador de EB (Ponte) e BB (CO).</p> <p>Acções iniciais:</p> <p>(1) Estabeleça comunicações com o CO/Ponte através do circuito 8 e solicite autorização para iniciar o SOC.</p> <p>(2) Em ambos os lançadores (BB e EB), retire as Protective Caps, coloque os Launcher Safety Switches em SAFE e volte a colocar as respectivas Protective Caps.</p> <p>AVISO: Certifique-se que a lâmpada do SAFETY SWITCH está acesa.</p> <p>(3) Retire e guarde as tampas de plástico de todos os tubos do lançador em teste.</p> <p>(4) Retire todas as munições que estejam eventualmente no lançador em teste e coloque-as por ordem no respectivo cofre de munições prontas.</p> <p>ATENÇÃO: Certifique-se que todos os tubos do lançador em teste estão vazios</p> <p>(5) No lançador em teste, retire de novo a Protective Cap, coloque o Launcher Safety Switch em ON e reinstale a Protective Cap.</p> <p>(6) Efectue o Reset ao indicador da Cartridge de teste e informe o CO/Ponte que está pronto a iniciar os testes.</p> <p>Procedimento:</p> <p>Teste de comando a partir do CO.</p> <p>Munição de CHAFF</p> <p>(1) À ordem do CO, coloque a Cartridge de teste no tubo 1.</p> <p>(2) Quando for simulado fogo a partir do CO, o ponteiro do indicador da Cartridge de teste deverá mover-se para a zona verde. Observe com atenção o indicador e, caso o ponteiro se mova para a zona verde, informe o CO: «Teste tubo 1 OK». Caso isso não se verifique, informe o CO: «Teste tubo 1 NEGATIVO».</p> <p>(3) Retire a Cartridge de teste do tubo e efectue o Reset ao indicador.</p>				

**GDH SOC'S**  
**REALIZADOS:** \_\_\_\_\_ **OPERADOR:** \_\_\_\_\_

## STANDARD OPERATOR CHECKS Operador de Sinais

**FFAH's Classe "Vasco da Gama"**

**SOC Nº 131**

**STANDARD OPERATOR CHECKS (SOC'S)**

**Equipamento: PROJECTOR DE SINAIS EB**

**Executante: OPERADOR SINAIS VISUAIS**

1. Desaperte os fixadores e verifique o movimento do projector no plano vertical e no plano horizontal. Ele roda sem problemas?
2. Accionando a alavanca a persiana abre bem?
3. Junto ao próprio projector accione o comutador de alimentação, para estabelecer o projector. Pressionando o botão de alimentação a lâmpada interior acende?

No caso do projector não estar alimentado solicite a verificação do disjuntor de alimentação localizado no painel 3 do quadro eléctrico 2-39 e repita a execução do passo 3.

Alinhe de novo o projector na posição correcta e desligue o comutador de alimentação.

SIM	NÃO

\*\*\*\*\* SOC CONCLUÍDO \*\*\*\*\*

**FFAH's Classe "Vasco da Gama"**

**SOC Nº 132**

**STANDARD OPERATOR CHECKS (SOC'S)**

**Equipamento: PROJECTOR DE SINAIS BB**

**Executante: OPERADOR SINAIS VISUAIS**

1. Desaperte os fixadores e verifique o movimento do projector no plano vertical e no plano horizontal. Ele roda sem problemas?
2. Accionando a alavanca a persiana abre bem?
3. Junto ao próprio projector accione o comutador de alimentação, para estabelecer o projector. Pressionando o botão de alimentação a lâmpada interior acende?

No caso do projector não estar alimentado solicite a verificação do disjuntor de alimentação localizado no painel 3 do quadro eléctrico 2-39 e repita a execução do passo 3.

Alinhe de novo o projector na posição correcta e desligue o comutador de alimentação.

SIM	NÃO

**\*\*\*\*\* SOC CONCLUÍDO \*\*\*\*\***

**FFAH's Classe "Vasco da Gama"**

**SOC Nº** \_\_\_\_

**STANDARD OPERATOR CHECKS (SOC'S)**

**Equipamento: ADRIÇAS E BALÕES**

**Executante: OPERADOR SINAIS VISUAIS**

1. No parque de sinais, todas as adriças estão em bom estado?
2. Todas as adriças correm bem nas respectivas roldanas?
3. Os gatos de cada adriça não apresentam sinais de desgaste?
4. Os balões de sinais estão disponíveis e devidamente peados?
5. A Bandeira Nacional de dois panos para usar com o navio no mar está em bom estado?
6. A flâmula está em bom estado?
7. O Jaque pronto a usar está em bom estado?

SIM	NÃO

**\*\*\*\*\* SOC CONCLUÍDO \*\*\*\*\***

**FFAH's Classe "Vasco da Gama"**

**SOC Nº 133**

**STANDARD OPERATOR CHECKS (SOC'S)**

**Equipamento: LANTERNA DE SINAIS PORTÁTIL**

**Executante: OPERADOR SINAIS VISUAIS**

Retire a lanterna da caixa respectiva e ligue o cabo de alimentação na ficha respectiva.

1. Verifique que o vidro e o espelho da óptica estão em boas condições?
2. Coloque o comutador ON/OFF na posição ON. Verifique que a lâmpada se acende?
3. Destrave o gatilho da lanterna. Accionando o gatilho o ocultador da lâmpada movimenta-se sem problemas?

Volte a colocar a lanterna na caixa.

SIM	NÃO

**\*\*\*\*\* SOC CONCLUÍDO \*\*\*\*\***

## FFAH's Classe "Vasco da Gama"

SOC Nº 134

### STANDARD OPERATOR CHECKS (SOC'S)

**Equipamento: LUZES OMNIDIRECCIONAIS**

**Executante: OPERADOR SINAIS VISUAIS**

No Painel de Alarmes e Segurança da Navegação (Ponte) coloque o selector de alimentação "I/II" para a posição "I"

**1.** Accionando o interruptor correspondente a "POSIÇÃO/SINAIS MORSE" os dois LEDs amarelos acendem não soando qualquer alarme sonoro?

**2.** Verifique no Mastro que as luzes se encontram acesas?

Desligue o interruptor correspondente a "POSIÇÃO/SINAIS MORSE".

**3.** Verifique com a colaboração dos Vigias que accionando as seguintes Chaves Morse as lâmpadas acendem:

**a.** Asa da Ponte de EB?

**b.** Asa da Ponte de BB?

**c.** Ponte (junto do Sealand 66)?

Desligue o comutador de alimentação do Painel de Alarmes e Segurança da Navegação, colocando-o na posição 0.

SIM	NÃO

\*\*\*\*\* SOC CONCLUÍDO \*\*\*\*\*

## Anexo C – Código fonte das *Activities* da Aplicação móvel

### MainActivity:

```
package tese.sistema.socs;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity {

    EditText Ip;
    Button Entrar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Ip = (EditText) findViewById (R.id.endereco_IP);
        Entrar = (Button ) findViewById (R.id.entrar);

        Entrar.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                new BuscarIP();
                Ip.getText().toString();

                BuscarIP.setValue(Ip.getText().toString());
                startActivity(new Intent("tese.sistema.Executantes"));

                // TODO Auto-generated method stub

            }
        });
    }

}
```

### Executantes:

```
package tese.sistema.socs;
```



```

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;

public class Executantes extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.executantes);

    }

    public void onClick1(View view){

        startActivity(new Intent("tese.sistema.OQP"));

    }

    public void onClick2(View view){

        startActivity(new Intent("tese.sistema.Sargento"));

    }

    public void onClick3(View view){

        startActivity(new Intent("tese.sistema.Marinheiro_Leme"));

    }

    public void onClick4(View view){

        startActivity(new Intent("tese.sistema.Operador_Sinais"));

    }

}

```

### OQP:

```

package tese.sistema.socs;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

```

```

public class OQP extends Activity{

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.oqp);
    }

    public void onClick1(View view){

        startActivity(new Intent("tese.sistema.KH_1007_NTD"));

    }

    public void onClick3(View view){
        startActivity(new Intent("tese.sistema.DGPS"));

    }

    public void onClick4(View view){
        startActivity(new Intent("tese.sistema.VHF_Sealand"));

    }

    public void onClick5(View view){
        startActivity(new Intent("tese.sistema.VHF_DSC_Sailor"));

    }

}

```

### Sargento:

```

package tese.sistema.socs;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class Sargento extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.sargento_oqp);

    }

    public void onClick2(View view){

```

```

        startActivity(new Intent("tese.sistema.Anemometro"));
    }

    public void onClick3(View view){
        startActivity(new Intent("tese.sistema.Luzes_Nav"));
    }

    public void onClick4(View view){
        startActivity(new Intent("tese.sistema.Sereia"));
    }
}

```

### **Marinheiro\_Leme:**

```

package tese.sistema.socs;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class Marinheiro_Leme extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.marinheiro_Leme);
    }

    public void onClick1(View view){
        startActivity(new Intent("tese.sistema.Angulo_Leme"));
    }

    public void onClick2(View view){
        startActivity(new Intent("tese.sistema.Piloto_Automatico"));
    }
}

```

### **Operador\_Sinais:**

```

package tese.sistema.socs;

```

```

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class Operador_Sinais extends Activity{

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.operador_sinais);
    }

    public void onClick1(View view){

        startActivity(new Intent("tese.sistema.Projector_Sinais_EB"));

    }

    public void onClick2(View view){
        startActivity(new
Intent("tese.sistema.Projector_Sinais_BB"));

        }

    public void onClick3(View view){
        startActivity(new Intent("tese.sistema.Adrica_Baloes"));

        }

    public void onClick4(View view){
        startActivity(new Intent("tese.sistema.Lanterna"));

    }

    public void onClick5(View view){
        startActivity(new Intent("tese.sistema.Luzes_Omni"));

    }

}

```

### **Adrica\_Baloes:**

```

package tese.sistema.socs;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;

```

```

import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Chronometer;
import android.widget.Toast;

public class Adrica_Baloes extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, item5, item6, item7, equipamento,
duracao;
    String[] estadoAdrica, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {

        estadoAdrica= new String[] {};
        super.onCreate(savedInstanceState);
        setContentView(R.layout.adrica_baloes);
        mChronometer = (Chronometer) findViewById(R.id.chronometer7);
        mChronometer.start();

        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        //Verificar Checkbox

        CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
        checkBox1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub

                if(((CheckBox)v).isChecked())
                {

                    item1 = "Funciona";

                }
                else
                {
                    item1 ="Falhou";
                }

            }

        });

        if (item1 == null)
            item1 = "Falhou";

        CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);

```

```

checkBox2.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item2 = "Funciona";

        }
        else
        {
            item2 = "Falhou";
        }
    }

});

if (item2 == null)
    item2 = "Falhou";

CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);

checkBox3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item3 = "Funciona";

        }
        else
        {
            item3 = "Falhou";
        }
    }

});

if (item3 == null)
    item3 = "Falhou";

```

```

CheckBox checkBox4 = (CheckBox) findViewById(R.id.checkBox4);

checkBox4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item4 = "Funciona";

```

```

    }
    else
    {

        item4 ="Falhou";
    }

    }

});

if (item4 == null)
    item4 ="Falhou";

```

```

CheckBox checkBox5 = (CheckBox) findViewById(R.id.checkBox5);

checkBox5.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item5 ="Funciona";

        }
        else
        {

            item5 ="Falhou";
        }

    }

});

if (item5 == null)
    item5 ="Falhou";

```

```

CheckBox checkBox6 = (CheckBox) findViewById(R.id.checkBox6);

checkBox6.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item6 ="Funciona";

        }
        else
        {

```

```

        item6 = "Falhou";
    }

    });

    if (item6 == null)
        item6 = "Falhou";

    CheckBox checkBox7 = (CheckBox) findViewById(R.id.checkBox7);

    checkBox7.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub

            if(((CheckBox)v).isChecked())
            {

                item7 = "Funciona";

            }
            else
            {

                item7 = "Falhou";
            }

        }

    });

    if (item7 == null)
        item7 = "Falhou";

    Button Terminar = (Button) findViewById(R.id.terminar);
    Terminar.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

            // TODO Auto-generated method stub
            mChronometer.stop();
            duracao = mChronometer.getText().toString();
            Estado_soc = GetEstado();
            equipamento = "Adrica";

            DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

            for(int i =0; i<Estado_soc.length; i++)
                AtualizarEstado(Estado_soc[i]);

            AtualizarRegisto(duracao, equipamento);
            DisplayToast("Os dados da SOC foram guardados na BD");
        }

    });

```



```

    });
}

    public void AtualizarEstado (String EstadoEquipamento){
        String Resultado = "";

        try {
            urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/GravarAdrica.jsp?Estado="+EstadoEquipa
mento;
            Resultado = ConectarHTTP.executarHttpGet(urlGet);
            Log.i("conexão" , "resultado = " +Resultado);

        }

        catch (Exception erro){
            Log.i("erro" , "erro = " +erro);
            erro.printStackTrace();
        }

    }

    public void AtualizarRegistro (String equipamento, String duracao ){
        String Resultado = "";

        try {
            urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/Registrar_tempo.jsp?Duracao="+duracao+"
&Equipamento="+equipamento;
            Resultado = ConectarHTTP.executarHttpGet(urlGet);
            Log.i("conexão" , "resultado = " +Resultado);

        }

        catch (Exception erro){
            Log.i("erro" , "erro = " +erro);
            erro.printStackTrace();
        }
    }
}

```

```

        }

    }

    public String[] GetEstado()
    {

        return estadoAdrica= new String[] {item1, item2, item3, item4, item5,
item6, item7};

    }

    private void DisplayToast(String mensagem) {

        // TODO Auto-generated method stub
        Toast.makeText(getBaseContext(), mensagem,
Toast.LENGTH_SHORT).show();

    }

}

```

### Anemometro:

```

package tese.sistema.socs;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Chronometer;
import android.widget.Toast;

public class Anemometro extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, equipamento, duracao;
    String[] estadoAnemometro, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        estadoAnemometro= new String[] {};
        super.onCreate(savedInstanceState);
        setContentView(R.layout.anemometro);
        mChronometer = (Chronometer) findViewById(R.id.chronometer);
        mChronometer.start();
    }
}

```

```

StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);

```

//Verificar Checkbox

```

CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
checkBox1.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item1 = "Funciona";

        }
        else
        {
            item1 ="Falhou";
        }

    }

});

if (item1 == null)
    item1 = "Falhou";

CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);

checkBox2.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item2 ="Funciona";

        }
        else
        {
            item2 ="Falhou";
        }

    }

});

if (item2 == null)
    item2 ="Falhou";

CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);

```

```

checkBox3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if((CheckBox)v.isChecked())
        {
            item3 ="Funciona";

        }
        else
        {

            item3 ="Falhou";
        }

    }

});

if (item3 == null)
    item3 ="Falhou";

```

```

CheckBox checkBox4 = (CheckBox) findViewById(R.id.checkBox4);

```

```

checkBox4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if((CheckBox)v.isChecked())
        {

            item4 ="Funciona";

        }
        else
        {

            item4 ="Falhou";
        }

    }

});

if (item4 == null)
    item4 ="Falhou";

```

```

Button Terminar = (Button) findViewById(R.id.terminar);
Terminar.setOnClickListener(new View.OnClickListener() {

```

```

    @Override
    public void onClick(View v) {

```

```

        // TODO Auto-generated method stub
        mChronometer.stop();
        duracao = mChronometer.getText().toString();
        Estado_soc = GetEstado();
        equipamento = "Anemometro";

        DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

        for(int i =0; i<Estado_soc.length; i++)
            ActualizarEstado(Estado_soc[i]);

        ActualizarRegisto(duracao, equipamento);
        DisplayToast("Os dados da SOC foram guardados na BD");

    }

});

}

public void ActualizarEstado (String EstadoEquipamento){
    String Resultado = "";

    try {
        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/GravarAnemometro.jsp?Estado="+EstadoEq
uipamento;
        Resultado = ConectarHTTP.executarHttpGet(urlGet);
        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){
        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();
    }

}

public void ActualizarRegisto (String equipamento, String duracao ){
    String Resultado = "";

```

```

        try {
            urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/Registrar_tempo.jsp?Duracao="+duracao+
&Equipamento="+equipamento;
            Resultado = ConectarHTTP.executarHttpGet(urlGet);

            Log.i("conexão" , "resultado = " +Resultado);
        }

        catch (Exception erro){

            Log.i("erro" , "erro = " +erro);

            erro.printStackTrace();

        }

    }

    public String[] GetEstado()
    {

        return estadoAnemometro= new String[] {item1, item2, item3, item4};

    }

    private void DisplayToast(String mensagem) {

        // TODO Auto-generated method stub
        Toast.makeText(getApplicationContext(), mensagem,
        Toast.LENGTH_SHORT).show();

    }

}

```

### Angulo\_Leme:

```

package tese.sistema.socs;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Chronometer;
import android.widget.Toast;

```

```

public class Angulo_Leme extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, item5, item6, item7, item8, item9,
    item10, item11, item12, item13,equipamento, duracao;
    String[] estadoAngulo, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        estadoAngulo= new String[] {};
        super.onCreate(savedInstanceState);
        setContentView(R.layout.angulo_Leme);
        mChronometer = (Chronometer) findViewById(R.id.chronometer4);
        mChronometer.start();

        StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        //Verificar Checkbox

        CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
        checkBox1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub

                if(((CheckBox)v).isChecked())
                {

                    item1 = "Funciona";

                }
                else
                {
                    item1 ="Falhou";
                }
            }

        });

        if (item1 == null)
            item1 = "Falhou";

        CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);

        checkBox2.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub

```

```

        if(((CheckBox)v).isChecked())
        {

            item2 ="Funciona";

        }
        else
        {
            item2 ="Falhou";
        }

    }

});

if (item2 == null)
    item2 ="Falhou";

CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);

checkBox3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item3 ="Funciona";

        }
        else
        {
            item3 ="Falhou";
        }

    }

});

if (item3 == null)
    item3 ="Falhou";

```

```

CheckBox checkBox4 = (CheckBox) findViewById(R.id.checkBox4);

checkBox4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item4 ="Funciona";

        }

    }

});

```



```

        else
        {
            item4 = "Falhou";
        }
    }
});

if (item4 == null)
    item4 = "Falhou";

```

```

CheckBox checkBox5 = (CheckBox) findViewById(R.id.checkBox5);

checkBox5.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item5 = "Funciona";

        }
        else
        {

            item5 = "Falhou";
        }

    }

});

if (item5 == null)
    item5 = "Falhou";

```

```

CheckBox checkBox6 = (CheckBox) findViewById(R.id.checkBox6);

checkBox6.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item6 = "Funciona";

        }
        else
        {

            item6 = "Falhou";
        }

    }

});

if (item6 == null)
    item6 = "Falhou";

```

```

        }
    }
});

if (item6 == null)
    item6 = "Falhou";

CheckBox checkBox7 = (CheckBox) findViewById(R.id.checkBox7);

checkBox7.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item7 = "Funciona";

        }
        else
        {

            item7 = "Falhou";
        }

    }

});

if (item7 == null)
    item7 = "Falhou";

CheckBox checkBox8 = (CheckBox) findViewById(R.id.checkBox8);

checkBox8.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item8 = "Funciona";

        }
        else
        {

            item8 = "Falhou";
        }

    }

});

```

```

        if (item8 == null)
            item8 = "Falhou";

CheckBox checkBox9 = (CheckBox) findViewById(R.id.checkBox9);

checkBox9.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item9 = "Funciona";

        }
        else
        {
            item9 = "Falhou";
        }

    }

});

if (item9 == null)
    item9 = "Falhou";

CheckBox checkBox10 = (CheckBox) findViewById(R.id.checkBox10);

checkBox10.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item10 = "Funciona";

        }
        else
        {
            item10 = "Falhou";
        }

    }

});

if (item10 == null)
    item10 = "Falhou";

CheckBox checkBox11 = (CheckBox) findViewById(R.id.checkBox11);

```

```

checkBox11.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item11 = "Funciona";

        }
        else
        {

            item11 = "Falhou";
        }

    }

});

if (item11 == null)
    item11 = "Falhou";

CheckBox checkBox12 = (CheckBox) findViewById(R.id.checkBox12);

checkBox12.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item12 = "Funciona";

        }
        else
        {

            item12 = "Falhou";
        }

    }

});

if (item12 == null)
    item12 = "Falhou";

CheckBox checkBox13 = (CheckBox) findViewById(R.id.checkBox13);

checkBox13.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

```

```

        if(((CheckBox)v).isChecked())
        {

            item13 ="Funciona";

        }
        else
        {

            item13 ="Falhou";
        }

    }

});

if (item13 == null)
    item13 ="Falhou";

    Button Terminar = (Button) findViewById(R.id.terminar);
    Terminar.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

            // TODO Auto-generated method stub
            mChronometer.stop();
            duracao = mChronometer.getText().toString();
            Estado_soc = GetEstado();
            equipamento = "Angulo_Leme";

            DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

            for(int i =0; i<Estado_soc.length; i++)
                ActualizarEstado(Estado_soc[i]);

            ActualizarRegisto(duracao, equipamento);
            DisplayToast("Os dados da SOC foram guardados na BD");

        }

    });

}

public void ActualizarEstado (String EstadoEquipamento){

    String Resultado = "";

    try {

```

```

        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/GravarAngulo.jsp?Estado="+EstadoEquipa
mento;

        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();

    }

}

public void AtualizarRegisto (String equipamento, String duracao ){

    String Resultado = "";

    try {

        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/Registrar_tempo.jsp?Duracao="+duracao+"
&Equipamento="+equipamento;
        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();

    }

}

public String[] GetEstado()
{

    return estadoAngulo= new String[] {item1, item2, item3, item4,
item5, item6, item7, item8, item9, item10, item11, item12, item13};

}

private void DisplayToast(String mensagem) {

```

```

        // TODO Auto-generated method stub
        Toast.makeText(getApplicationContext(), mensagem,
        Toast.LENGTH_SHORT).show();

    }

}

```

## DGPS:

```

package tese.sistema.socs;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Chronometer;
import android.widget.Toast;

public class DGPS extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, item5, item6, equipamento, duracao;
    String[] estadoDgps, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        estadoDgps= new String[] {};
        super.onCreate(savedInstanceState);
        setContentView(R.layout.dgps_furuno);
        mChronometer = (Chronometer) findViewById(R.id.chronometer);
        mChronometer.start();

        StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        //Verificar Checkbox

        CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
        checkBox1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub

                if(((CheckBox)v).isChecked())
                {

```

```

        item1 = "Funciona";
    }
    else
    {
        item1 = "Falhou";
    }
    }
});

if (item1 == null)
    item1 = "Falhou";

CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);

checkBox2.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item2 = "Funciona";

        }
        else
        {
            item2 = "Falhou";
        }

    }

});

if (item2 == null)
    item2 = "Falhou";

CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);

checkBox3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item3 = "Funciona";

        }
        else
        {
            item3 = "Falhou";
        }

    }

});

```



```

    }
});

if (item3 == null)
    item3 = "Falhou";

CheckBox checkBox4 = (CheckBox) findViewById(R.id.checkBox4);

checkBox4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item4 = "Funciona";

        }
        else
        {

            item4 = "Falhou";
        }

    }

});

if (item4 == null)
    item4 = "Falhou";

CheckBox checkBox5 = (CheckBox) findViewById(R.id.checkBox5);

checkBox5.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item5 = "Funciona";

        }
        else
        {

            item5 = "Falhou";
        }

    }

});

```

```

    }
});

if (item5 == null)
    item5 = "Falhou"

CheckBox checkBox6 = (CheckBox) findViewById(R.id.checkBox6);

checkBox6.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item6 = "Funciona";

        }
        else
        {

            item6 = "Falhou";
        }

    }

});

if (item6 == null)
    item6 = "Falhou";

Button Terminar = (Button) findViewById(R.id.terminar);
Terminar.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

        // TODO Auto-generated method stub
        mChronometer.stop();
        duracao = mChronometer.getText().toString();
        Estado_soc = GetEstado();
        equipamento = "DGPS";
        DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

        for(int i =0; i<Estado_soc.length; i++)
            ActualizarEstado(Estado_soc[i]);

        ActualizarRegisto(duracao, equipamento);
        DisplayToast("Os dados da SOC foram guardados na BD");

    }

}

```

```

    });

}

    public void AtualizarEstado (String EstadoEquipamento){

        String Resultado = "";

        try {

            urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/GravarDGPS.jsp?Estado="+EstadoEquipame
nto;

            Resultado = ConectarHTTP.executarHttpGet(urlGet);

            Log.i("conexão" , "resultado = " +Resultado);

        }

        catch (Exception erro){

            Log.i("erro" , "erro = " +erro);

            erro.printStackTrace();

        }

    }

    public void AtualizarRegisto (String equipamento, String duracao ){

        String Resultado = "";

        try {

            urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/Registrar_tempo.jsp?Duracao="+duracao+"
&Equipamento="+equipamento;

            Resultado = ConectarHTTP.executarHttpGet(urlGet);

            Log.i("conexão" , "resultado = " +Resultado);

        }

        catch (Exception erro){

            Log.i("erro" , "erro = " +erro);

```

```

        erro.printStackTrace();
    }

}

public String[] GetEstado()
{

    return estadoDgps= new String[] {item1, item2, item3, item4, item5,
item6};

}

private void DisplayToast(String mensagem) {

    // TODO Auto-generated method stub
    Toast.makeText(getBaseContext(), mensagem,
Toast.LENGTH_LONG).show();

}

```

### KH\_1007\_NTD:

```

package tese.sistema.socs;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Chronometer;
import android.widget.Toast;

public class KH_1007_NTD extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, item5, item6, item7, item8, item9,
equipamento, duracao;
    String[] estadoKH, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        estadoKH= new String[] {};
        super.onCreate(savedInstanceState);
        setContentView(R.layout.kh_1007_ntd);
    }
}

```

```

        mChronometer = (Chronometer) findViewById(R.id.chronometer7);
        mChronometer.start();

        StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

```

#### //Verificar Checkbox

```

CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
checkBox1.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item1 = "Funciona";

        }
        else
        {
            item1 ="Falhou";
        }

    }

});

if (item1 == null)
    item1 = "Falhou";

CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);

checkBox2.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item2 ="Funciona";

        }
        else
        {
            item2 ="Falhou";
        }

    }

});

if (item2 == null)
    item2 ="Falhou";

```

```

CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);

checkBox3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item3 = "Funciona";

        }
        else
        {

            item3 = "Falhou";
        }

    }

});

if (item3 == null)
    item3 = "Falhou";

```

```

CheckBox checkBox4 = (CheckBox) findViewById(R.id.checkBox4);

checkBox4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item4 = "Funciona";

        }
        else
        {

            item4 = "Falhou";
        }

    }

});

if (item4 == null)
    item4 = "Falhou";

```

```

CheckBox checkBox5 = (CheckBox) findViewById(R.id.checkBox5);

checkBox5.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View v) {
    // TODO Auto-generated method stub

    if(((CheckBox)v).isChecked())
    {

        item5 ="Funciona";

    }
    else
    {

        item5 ="Falhou";
    }

}

});

if (item5 == null)
    item5 ="Falhou";

```

```

CheckBox checkBox6 = (CheckBox) findViewById(R.id.checkBox6);

```

```

checkBox6.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item6 ="Funciona";

        }
        else
        {

            item6 ="Falhou";
        }

    }

});

if (item6 == null)
    item6 ="Falhou";

```

```

CheckBox checkBox7 = (CheckBox) findViewById(R.id.checkBox7);

```

```

checkBox7.setOnClickListener(new View.OnClickListener() {

    @Override

```

```

        public void onClick(View v) {
            // TODO Auto-generated method stub

            if(((CheckBox)v).isChecked())
            {

                item7 ="Funciona";

            }
            else
            {

                item7 ="Falhou";
            }

        }
    });

    if (item7 == null)
        item7 ="Falhou";

```

```

CheckBox checkBox8 = (CheckBox) findViewById(R.id.checkBox8);

checkBox8.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item8 ="Funciona";

        }
        else
        {

            item8 ="Falhou";
        }

    }

});

if (item8 == null)
    item8 ="Falhou";

```

```

CheckBox checkBox9 = (CheckBox) findViewById(R.id.checkBox9);

checkBox9.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())

```



```

        {
            item9 ="Funciona";
        }
        else
        {
            item9 ="Falhou";
        }
    }
});

if (item9 == null)
    item9 ="Falhou";

    Button Terminar = (Button) findViewById(R.id.terminar);
    Terminar.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

            // TODO Auto-generated method stub
            mChronometer.stop();
            duracao = mChronometer.getText().toString();
            Estado_soc = GetEstado();
            equipamento = "KH_1007_NTD";
            DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

            for(int i =0; i<Estado_soc.length; i++)
                ActualizarEstado(Estado_soc[i]);

            ActualizarRegisto(duracao, equipamento);
            DisplayToast("Os dados da SOC foram guardados na BD");

        }

    });

}

public void ActualizarEstado (String EstadoEquipamento){

    String Resultado = "";

    try {

```

```

        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/GravarKH.jsp?Estado="+EstadoEquipament
o;

        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();

    }

}

public void AtualizarRegisto (String equipamento, String duracao ){

    String Resultado = "";

    try {

        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/Registrar_tempo.jsp?Duracao="+duracao+"
&Equipamento="+equipamento;
        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();

    }

}

public String[] GetEstado()
{

    return estadoKH= new String[] {item1, item2, item3, item4, item5,
item6, item7, item8, item9};

}

```

```

        private void DisplayToast(String mensagem) {

            // TODO Auto-generated method stub
            Toast.makeText(getApplicationContext(), mensagem,
            Toast.LENGTH_LONG).show();

        }

```

## Lanterna :

```

package tese.sistema.socs;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;

import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Chronometer;
import android.widget.Toast;

public class Lanterna extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, item5, equipamento, duracao;
    String[] estadoLanterna, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {

        estadoLanterna= new String[] {};
        super.onCreate(savedInstanceState);
        setContentView(R.layout.Lanterna);
        mChronometer = (Chronometer) findViewById(R.id.chronometer8);
        mChronometer.start();

        StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        //Verificar Checkbox

        CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
        checkBox1.setOnClickListener(new View.OnClickListener() {

            @Override

```

```

        public void onClick(View v) {
            // TODO Auto-generated method stub

            if(((CheckBox)v).isChecked())
            {

                item1 = "Funciona";

            }
            else
            {
                item1 ="Falhou";
            }

        }

    });

    if (item1 == null)
        item1 = "Falhou";

    CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);

    checkBox2.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub

            if(((CheckBox)v).isChecked())
            {

                item2 ="Funciona";

            }
            else
            {
                item2 ="Falhou";
            }

        }

    });

    if (item2 == null)
        item2 ="Falhou";

    CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);

    checkBox3.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub

```

```

        if(((CheckBox)v).isChecked())
        {

            item3 ="Funciona";

        }
        else
        {

            item3 ="Falhou";

        }

    }

});

if (item3 == null)
    item3 ="Falhou";

```

```

CheckBox checkBox4 = (CheckBox) findViewById(R.id.checkBox4);

checkBox4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item4 ="Funciona";

        }
        else
        {

            item4 ="Falhou";

        }

    }

});

if (item4 == null)
    item4 ="Falhou";

```

```

CheckBox checkBox5 = (CheckBox) findViewById(R.id.checkBox5);

checkBox5.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

```

```

        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item5 ="Funciona";

        }
        else
        {

            item5 ="Falhou";

        }

    }

});

if (item5 == null)
    item5 ="Falhou";

    Button Terminar = (Button) findViewById(R.id.terminar);
    Terminar.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

            // TODO Auto-generated method stub
            mChronometer.stop();
            duracao = mChronometer.getText().toString();
            Estado_soc = GetEstado();
            equipamento = "Lanterna";
            DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

            for(int i =0; i<Estado_soc.length; i++)
                ActualizarEstado(Estado_soc[i]);

            ActualizarRegisto(duracao, equipamento);
            DisplayToast("Os dados da SOC foram guardados na BD");

        }

    });

}

public void ActualizarEstado (String EstadoEquipamento){

```

```

String Resultado = "";

    try {

        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/GravarLanterna.jsp?Estado="+EstadoEqui
pamento;

        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();

    }

}

public void ActualizarRegistro (String equipamento, String duracao ){

    String Resultado = "";

    try {

        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/Registrar_tempo.jsp?Duracao="+duracao+"
&Equipamento="+equipamento;

        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();

    }

}

```

```

    public String[] GetEstado()
    {

        return estadolanterna= new String[] {item1, item2, item3, item4,
item5};

    }

    private void DisplayToast(String mensagem) {

        // TODO Auto-generated method stub
        Toast.makeText(getApplicationContext(), mensagem,
Toast.LENGTH_SHORT).show();

    }

}

```

### **Luzes\_Nav:**

```

package tese.sistema.socs;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Chronometer;
import android.widget.Toast;

public class Luzes_Nav extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, item5, item6, equipamento, duracao;
    String[] estadoLuzes, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        estadoLuzes= new String[] {};
        super.onCreate(savedInstanceState);
    }

```



```

        setContentView(R.layout.luzes_nav);
        mChronometer = (Chronometer) findViewById(R.id.chronometer9);
        mChronometer.start();

        StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

```

//Verificar Checkbox

```

CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
checkBox1.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if((CheckBox)v.isChecked())
        {

            item1 = "Funciona";

        }
        else
        {
            item1 = "Falhou";
        }

    }

});

if (item1 == null)
    item1 = "Falhou";

CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);

checkBox2.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if((CheckBox)v.isChecked())
        {

            item2 = "Funciona";

        }
        else
        {
            item2 = "Falhou";
        }

    }

});

```

```

    }
});

if (item2 == null)
    item2 = "Falhou";

CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);

checkBox3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item3 = "Funciona";

        }
        else
        {

            item3 = "Falhou";
        }

    }

});

if (item3 == null)
    item3 = "Falhou";

```

```

CheckBox checkBox4 = (CheckBox) findViewById(R.id.checkBox4);

checkBox4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item4 = "Funciona";

        }
        else
        {

            item4 = "Falhou";
        }

    }

});

```

```

    }
});

if (item4 == null)
    item4 = "Falhou";

```

```

CheckBox checkBox5 = (CheckBox) findViewById(R.id.checkBox5);

checkBox5.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item5 = "Funciona";

        }
        else
        {

            item5 = "Falhou";
        }

    }

});

if (item5 == null)
    item5 = "Falhou";

```

```

CheckBox checkBox6 = (CheckBox) findViewById(R.id.checkBox6);

checkBox6.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item6 = "Funciona";

        }
        else
        {

```

```

        item6 = "Falhou";
    }

    });

    if (item6 == null)
        item6 = "Falhou";

    Button Terminar = (Button) findViewById(R.id.terminar);
    Terminar.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

            // TODO Auto-generated method stub
            mChronometer.stop();
            duracao = mChronometer.getText().toString();
            Estado_soc = GetEstado();
            equipamento = "Luzes";
            DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

            for(int i =0; i<Estado_soc.length; i++)
                ActualizarEstado(Estado_soc[i]);

            ActualizarRegisto(duracao, equipamento);
            DisplayToast("Os dados da SOC foram guardados na BD");

        }

    });

}

public void ActualizarEstado (String EstadoEquipamento){

    String Resultado = "";

    try {

        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/GravarLuzesNav.jsp?Estado="+EstadoEqui
pamento;

```

```

        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();

    }

}

public void ActualizarRegisto (String equipamento, String duracao ){

    String Resultado = "";

    try {

        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/Registrar_tempo.jsp?Duracao="+duracao+"
&Equipamento="+equipamento;
        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();

    }

}

public String[] GetEstado()
{

    return estadoLuzes= new String[] {item1, item2, item3, item4,
item5, item6};

}

```

```

        private void DisplayToast(String mensagem) {

            // TODO Auto-generated method stub
            Toast.makeText(getApplicationContext(), mensagem,
            Toast.LENGTH_SHORT).show();

        }

    }
}

```

## Luzes\_Omni:

```

package tese.sistema.socs;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Chronometer;
import android.widget.Toast;

public class Luzes_Omni extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, item5, item6, item7, item8, item9,
    equipamento, duracao;
    String[] estadoLuzes, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        estadoLuzes= new String[] {};
        super.onCreate(savedInstanceState);
        setContentView(R.layout.luzes_omni);
        mChronometer = (Chronometer) findViewById(R.id.chronometer9);
        mChronometer.start();

        StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
    }
}

```

//Verificar Checkbox

```
CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
checkBox1.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item1 = "Funciona";

        }
        else
        {
            item1 ="Falhou";
        }

    }

});

if (item1 == null)
    item1 = "Falhou";

CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);

checkBox2.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item2 ="Funciona";

        }
        else
        {
            item2 ="Falhou";
        }

    }

});

if (item2 == null)
    item2 ="Falhou";
```

```

CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);

checkBox3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item3 = "Funciona";

        }
        else
        {

            item3 = "Falhou";
        }

    }

});

if (item3 == null)
    item3 = "Falhou";

```

```

CheckBox checkBox4 = (CheckBox) findViewById(R.id.checkBox4);

checkBox4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item4 = "Funciona";

        }
        else
        {

            item4 = "Falhou";
        }

    }

});

if (item4 == null)
    item4 = "Falhou";

```



```

CheckBox checkBox5 = (CheckBox) findViewById(R.id.checkBox5);

checkBox5.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item5 = "Funciona";

        }
        else
        {

            item5 = "Falhou";
        }

    }

});

if (item5 == null)
    item5 = "Falhou";

```

```

CheckBox checkBox6 = (CheckBox) findViewById(R.id.checkBox6);

checkBox6.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item6 = "Funciona";

        }
        else
        {

            item6 = "Falhou";
        }

    }

});

```

```

if (item6 == null)
    item6 = "Falhou";

```

```

CheckBox checkBox7 = (CheckBox) findViewById(R.id.checkBox7);

```

```

checkBox7.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item7 = "Funciona";

        }
        else
        {

            item7 = "Falhou";
        }

    }

});

```

```

if (item7 == null)
    item7 = "Falhou";

```

```

CheckBox checkBox8 = (CheckBox) findViewById(R.id.checkBox8);

```

```

checkBox8.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item8 = "Funciona";

        }
        else
        {

            item8 = "Falhou";
        }

    }

});

```

```

        if (item8 == null)
            item8 = "Falhou";

CheckBox checkBox9 = (CheckBox) findViewById(R.id.checkBox9);

checkBox9.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item9 = "Funciona";

        }
        else
        {

            item9 = "Falhou";
        }

    }

});

if (item9 == null)
    item9 = "Falhou";

Button Terminar = (Button) findViewById(R.id.terminar);
Terminar.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

        // TODO Auto-generated method stub
        mChronometer.stop();
        duracao = mChronometer.getText().toString();
        Estado_soc = GetEstado();
        equipamento = "Luzes_Omni";
        DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

        for(int i =0; i<Estado_soc.length; i++)
            ActualizarEstado(Estado_soc[i]);

        ActualizarRegisto(duracao, equipamento);
        DisplayToast("Os dados da SOC foram guardados na BD");

    }

});

```

```

}

    public void AtualizarEstado (String EstadoEquipamento){

        String Resultado = "";

        try {

            urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/GravarLuzes.jsp?Estado="+EstadoEquipam
ento;
            Resultado = ConectarHTTP.executarHttpGet(urlGet);

            Log.i("conexão" , "resultado = " +Resultado);

        }

        catch (Exception erro){

            Log.i("erro" , "erro = " +erro);

            erro.printStackTrace();

        }

    }

    public void AtualizarRegistro (String equipamento, String duracao ){

        String Resultado = "";

        try {

            urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/Registrar_tempo.jsp?Duracao="+duracao+"
&Equipamento="+equipamento;
            Resultado = ConectarHTTP.executarHttpGet(urlGet);

            Log.i("conexão" , "resultado = " +Resultado);

        }

        catch (Exception erro){

            Log.i("erro" , "erro = " +erro);

            erro.printStackTrace();

        }

    }

```

```

    }

    public String[] GetEstado()
    {
        return estadoLuzes= new String[] {item1, item2, item3, item4,
item5, item6, item7, item8, item9};
    }

    private void DisplayToast(String mensagem) {

        // TODO Auto-generated method stub
        Toast.makeText(getBaseContext(), mensagem,
Toast.LENGTH_SHORT).show();

    }

}

```

### **Piloto\_Automatico:**

```

package tese.sistema.socs;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Chronometer;
import android.widget.Toast;

public class Piloto_Automatico extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, item5, item6, item7, item8, item9,
item10, item11, item12, equipamento, duracao;
    String[] estadoPiloto, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        estadoPiloto= new String[] {};
        super.onCreate(savedInstanceState);
        setContentView(R.layout.piloto_automatico);
    }
}

```

```

        mChronometer = (Chronometer) findViewById(R.id.chronometer);
        mChronometer.start();

        StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

//Verificar Checkbox

        CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
        checkBox1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub

                if(((CheckBox)v).isChecked())
                {
                    item1 = "Funciona";

                }
                else
                {
                    item1 = "Falhou";
                }

            }

        });

        if (item1 == null)
            item1 = "Falhou";

        CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);

        checkBox2.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub

                if(((CheckBox)v).isChecked())
                {
                    item2 = "Funciona";

                }
                else
                {
                    item2 = "Falhou";
                }

            }

        });

        if (item2 == null)
            item2 = "Falhou";

        CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);

```

```

checkBox3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item3 ="Funciona";

        }
        else
        {

            item3 ="Falhou";
        }

    }

});

if (item3 == null)
    item3 ="Falhou";

```

```

CheckBox checkBox4 = (CheckBox) findViewById(R.id.checkBox4);

```

```

checkBox4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item4 ="Funciona";

        }
        else
        {

            item4 ="Falhou";
        }

    }

});

if (item4 == null)
    item4 ="Falhou";

```

```

CheckBox checkBox5 = (CheckBox) findViewById(R.id.checkBox5);

```

```

checkBox5.setOnClickListener(new View.OnClickListener() {

    @Override

```

```

        public void onClick(View v) {
            // TODO Auto-generated method stub

            if(((CheckBox)v).isChecked())
            {
                item5 ="Funciona";
            }
            else
            {
                item5 ="Falhou";
            }
        }
    });

    if (item5 == null)
        item5 ="Falhou";

```

```

CheckBox checkBox6 = (CheckBox) findViewById(R.id.checkBox6);

checkBox6.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item6 ="Funciona";

        }
        else
        {
            item6 ="Falhou";
        }
    }

});

if (item6 == null)
    item6 ="Falhou";

```

```

CheckBox checkBox7 = (CheckBox) findViewById(R.id.checkBox7);

checkBox7.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

```



```

        if(((CheckBox)v).isChecked())
        {

            item7 ="Funciona";

        }
        else
        {

            item7 ="Falhou";

        }

    }

});

if (item7 == null)
    item7 ="Falhou";

CheckBox checkBox8 = (CheckBox) findViewById(R.id.checkBox8);

checkBox8.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item8 ="Funciona";

        }
        else
        {

            item8 ="Falhou";

        }

    }

});

if (item8 == null)
    item8 ="Falhou";

CheckBox checkBox9 = (CheckBox) findViewById(R.id.checkBox9);

checkBox9.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item9 ="Funciona";

```

```

    }
    else
    {
        item9 = "Falhou";
    }
}
});

if (item9 == null)
    item9 = "Falhou";

```

```

CheckBox checkBox10 = (CheckBox) findViewById(R.id.checkBox10);

```

```

checkBox10.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item10 = "Funciona";

        }
        else
        {
            item10 = "Falhou";
        }
    }
});

if (item10 == null)
    item10 = "Falhou";

```

```

CheckBox checkBox11 = (CheckBox) findViewById(R.id.checkBox11);

```

```

checkBox11.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item11 = "Funciona";

        }
        else
        {

```

```

        item11 = "Falhou";
    }
    });

    if (item11 == null)
        item11 = "Falhou";

    CheckBox checkBox12 = (CheckBox) findViewById(R.id.checkBox12);

    checkBox12.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub

            if(((CheckBox)v).isChecked())
            {

                item12 = "Funciona";

            }
            else
            {

                item12 = "Falhou";
            }

        }
    });

    if (item12 == null)
        item12 = "Falhou";

    Button Terminar = (Button) findViewById(R.id.terminar);
    Terminar.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

            mChronometer.stop();
            duracao = mChronometer.getText().toString();
            Estado_soc = GetEstado();
            equipamento = "Piloto_Automatico";
            DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

            for(int i =0; i<Estado_soc.length; i++)
                ActualizarEstado(Estado_soc[i]);

            ActualizarRegisto(duracao, equipamento);

```

```

        DisplayToast("Os dados da SOC foram guardados na BD");
    }

});

}

    public void ActualizarEstado (String EstadoEquipamento){

        String Resultado = "";
        try {

            urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/GravarPiloto.jsp?Estado="+EstadoEquipa
mento;

            Resultado = ConectarHTTP.executarHttpGet(urlGet);

            Log.i("conexão" , "resultado = " +Resultado);

        }

        catch (Exception erro){

            Log.i("erro" , "erro = " +erro);

            erro.printStackTrace();

        }

    }

    public void ActualizarRegisto (String equipamento, String duracao ){

        String Resultado = "";

        try {

            urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/Registar_tempo.jsp?Duracao="+duracao+"
&Equipamento="+equipamento;

            Resultado = ConectarHTTP.executarHttpGet(urlGet);

            Log.i("conexão" , "resultado = " +Resultado);

        }

        catch (Exception erro){

            Log.i("erro" , "erro = " +erro);

            erro.printStackTrace();

        }

    }

```

```

        }

    public String[] GetEstado()
    {

        return estadoPiloto= new String[] {item1, item2, item3, item4,
item5, item6, item7, item8, item9, item10, item11, item12};

    }

    private void DisplayToast(String mensagem) {

        // TODO Auto-generated method stub
        Toast.makeText(getApplicationContext(), mensagem,
Toast.LENGTH_SHORT).show();

    }

```

### Projector\_Sinais\_BB:

```

package tese.sistema.socs;
import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Chronometer;
import android.widget.Toast;

public class Projector_Sinais_BB extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, item5, equipamento, duracao;
    String[] estadoProjector, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.projector_sinais_bb);
        mChronometer = (Chronometer) findViewById(R.id.chronometer7);
        mChronometer.start();

        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();

```

```
StrictMode.setThreadPolicy(policy);
```

```
//Verificar Checkbox
```

```
CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
checkBox1.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item1 = "Funciona";

        }
        else
        {
            item1 ="Falhou";
        }

    }

});

if (item1 == null)
    item1 = "Falhou";

CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);
checkBox2.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item2 ="Funciona";

        }
        else
        {
            item2 ="Falhou";
        }

    }

});

if (item2 == null)
    item2 ="Falhou";

CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);
```

```

checkBox3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item3 = "Funciona";

        }
        else
        {

            item3 = "Falhou";
        }

    }

});

if (item3 == null)
    item3 = "Falhou";

CheckBox checkBox4 = (CheckBox) findViewById(R.id.checkBox4);

checkBox4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item4 = "Funciona";

        }
        else
        {

            item4 = "Falhou";
        }

    }

});

if (item4 == null)
    item4 = "Falhou";

```

```

CheckBox checkBox5 = (CheckBox) findViewById(R.id.checkBox5);

checkBox5.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

```

```

        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item5 ="Funciona";

        }
        else
        {

            item5 ="Falhou";

        }

    }

});

if (item5 == null)
    item5 ="Falhou";

    Button Terminar = (Button) findViewById(R.id.terminar);
    Terminar.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

            // TODO Auto-generated method stub
            mChronometer.stop();
            duracao = mChronometer.getText().toString();
            Estado_soc = GetEstado();
            equipamento = "Projector_Sinais_BB";
            DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

            for(int i =0; i<Estado_soc.length; i++)
                ActualizarEstado(Estado_soc[i]);

            ActualizarRegisto(duracao, equipamento);
            DisplayToast("Os dados da SOC foram guardados na BD");

        }

    });

}

public void ActualizarEstado (String EstadoEquipamento){

    String Resultado = "";

    try {

```



```

        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/GravarProjectorBB.jsp?Estado="+EstadoE
quipamento;

        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();

    }

}

public void AtualizarRegisto (String equipamento, String duracao ){

    String Resultado = "";

    try {

        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/Registrar_tempo.jsp?Duracao="+duracao+"
&Equipamento="+equipamento;
        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();

    }

}

public String[] GetEstado()
{

    return estadoProjector= new String[] {item1, item2, item3, item4,
item5};

}

private void DisplayToast(String mensagem) {

```

```

        // TODO Auto-generated method stub
        Toast.makeText(getApplicationContext(), mensagem,
        Toast.LENGTH_SHORT).show();

    }

}

```

## Projector\_Sinais\_EB

```

package tese.sistema.socs;
import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Chronometer;
import android.widget.Toast;

public class Projector_Sinais_EB extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, item5, duracao, equipamento;
    String[] estadoProjector, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.projector_sinais_eb);
        mChronometer = (Chronometer) findViewById(R.id.chronometer5);
        mChronometer.start();

        StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        //Verificar Checkbox

        CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
        checkBox1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub

                if(((CheckBox)v).isChecked())

```

```

        {

            item1 = "Funciona";

        }
        else
        {
            item1 = "Falhou";
        }

    }

});

if (item1 == null)
    item1 = "Falhou";

CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);

checkBox2.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item2 = "Funciona";

        }
        else
        {
            item2 = "Falhou";
        }

    }

});

if (item2 == null)
    item2 = "Falhou";

CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);

checkBox3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item3 = "Funciona";

```

```

        }
        else
        {

            item3 ="Falhou";
        }

    }

});

if (item3 == null)
    item3 ="Falhou";

```

```

CheckBox checkBox4 = (CheckBox) findViewById(R.id.checkBox4);

checkBox4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item4 ="Funciona";

        }
        else
        {

            item4 ="Falhou";
        }

    }

});

if (item4 == null)
    item4 ="Falhou";

```

```

CheckBox checkBox5 = (CheckBox) findViewById(R.id.checkBox5);

checkBox5.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item5 ="Funciona";

        }

    }

});

```

```

        else
        {
            item5 ="Falhou";
        }

    }

});

if (item5 == null)
    item5 ="Falhou";

    Button Terminar = (Button) findViewById(R.id.terminar);
    Terminar.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

            // TODO Auto-generated method stub
            mChronometer.stop();
            duracao = mChronometer.getText().toString();
            Estado_soc = GetEstado();
            equipamento = "Projector_Sinais_EB";
            DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

            for(int i =0; i<Estado_soc.length; i++)
                ActualizarEstado(Estado_soc[i]);

            ActualizarRegisto(duracao, equipamento);
            DisplayToast("Os dados da SOC foram guardados na BD");

        }

    });

}

public void ActualizarEstado (String EstadoEquipamento){

    String Resultado = "";

    try {

        urlGet = "http://
"+BuscarIP.getValue()+" :8080/SocsWebbApp/GravarProjectorEB.jsp?Estado="+EstadoE
quipamento;

        Resultado = ConectarHTTP.executarHttpGet(urlGet);
    }
}

```

```

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();

    }

}

public void ActualizarRegisto (String equipamento, String duracao ){

    String Resultado = "";

    try {

        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/Registrar_tempo.jsp?Duracao="+duracao+"
&Equipamento="+equipamento;
        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();

    }

}

public String[] GetEstado()
{

    return estadoProjector= new String[] {item1, item2, item3, item4,
item5};

}

private void DisplayToast(String mensagem) {

    // TODO Auto-generated method stub

```

```

        Toast.makeText(getBaseContext(), mensagem,
Toast.LENGTH_SHORT).show();

    }

}

```

### Sereia:

```

package tese.sistema.socs;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.Chronometer;
import android.widget.CheckBox;
import android.widget.Toast;
import android.widget.Button;

public class Sereia extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, equipamento, duracao;
    String[] estadoBuzina, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        estadoBuzina= new String[] {};
        super.onCreate(savedInstanceState);
        setContentView(R.layout.sereia);
        mChronometer = (Chronometer) findViewById(R.id.chronometer1);
        mChronometer.start();

        StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        //Verificar Checkbox

        CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
        checkBox1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub

                if(((CheckBox)v).isChecked())
                {
                    item1 = "Funciona";
                }
            }
        }
    }
}

```

```

        else
        {
            item1 = "Falhou";
        }
    }
});

if (item1 == null)
    item1 = "Falhou";

CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);

checkBox2.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item2 = "Funciona";
        }
        else
        {
            item2 = "Falhou";
        }
    }

});

if (item2 == null)
    item2 = "Falhou";

CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);

checkBox3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item3 = "Funciona";
        }
        else
        {
            item3 = "Falhou";
        }
    }

});

```



```

    });

    if (item3 == null)
        item3 = "Falhou";

    // Botão que deve ser clicado para terminar a SOC e para o
    Cronometro
    Button Terminar = (Button) findViewById(R.id.terminar);
    Terminar.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

            mChronometer.stop();
            duracao = mChronometer.getText().toString();
            Estado_soc = GetEstado();
            equipamento = "Sereia";
            DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

            for(int i =0; i<Estado_soc.length; i++)
                ActualizarEstado(Estado_soc[i]);

            ActualizarRegisto(duracao, equipamento);
            DisplayToast("Os dados da SOC foram guardados na BD");

        }

    });

}

public void ActualizarEstado (String EstadoEquipamento){

    String Resultado = "";

    try {

        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/GravarBuzina.jsp?Estado="+EstadoEquipa
mento;
        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();
    }
}

```

```

        }
    }

    public void ActualizarRegistro (String equipamento, String duracao ){

        String Resultado = "";

        try {
            urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/Registrar_tempo.jsp?Duracao="+duracao+"
&Equipamento="+equipamento;
            Resultado = ConectarHTTP.executarHttpGet(urlGet);

            Log.i("conexão" , "resultado = " +Resultado);

        }

        catch (Exception erro){

            Log.i("erro" , "erro = " +erro);

            erro.printStackTrace();

        }

    }

    public String[] GetEstado()
    {

        return estadoBuzina= new String[] {item1, item2, item3};

    }

    private void DisplayToast(String mensagem) {

        // TODO Auto-generated method stub
        Toast.makeText(getApplicationContext(), mensagem,
        Toast.LENGTH_LONG).show();

    }

```

## VHF\_DSC\_Sailor :

```

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;

```

```

import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Chronometer;
import android.widget.Toast;

public class VHF_DSC_Sailor extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, item5, item6, item7, item8, item9,
item10, equipamento, duracao;
    String[] estadoDsc, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        estadoDsc= new String[] {};
        super.onCreate(savedInstanceState);
        setContentView(R.layout.vhf_dsc_sailor);
        mChronometer = (Chronometer) findViewById(R.id.chronometer);
        mChronometer.start();

        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        //Verificar Checkbox

        CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
        checkBox1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub

                if(((CheckBox)v).isChecked())
                {
                    item1 = "Funciona";

                }
                else
                {
                    item1 ="Falhou";
                }

            }

        });

        if (item1 == null)
            item1 = "Falhou";

        CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);

```

```

checkBox2.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item2 = "Funciona";

        }
        else
        {
            item2 = "Falhou";
        }

    }

});

if (item2 == null)
    item2 = "Falhou";

CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);

checkBox3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item3 = "Funciona";

        }
        else
        {

            item3 = "Falhou";

        }

    }

});

if (item3 == null)
    item3 = "Falhou";

```

```

CheckBox checkBox4 = (CheckBox) findViewById(R.id.checkBox4);

```

```

checkBox4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())

```

```

        {
            item4 = "Funciona";
        }
        else
        {
            item4 = "Falhou";
        }
    }
});

if (item4 == null)
    item4 = "Falhou";

```

```

CheckBox checkBox5 = (CheckBox) findViewById(R.id.checkBox5);

checkBox5.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item5 = "Funciona";

        }
        else
        {
            item5 = "Falhou";
        }
    }

});

if (item5 == null)
    item5 = "Falhou";

```

```

CheckBox checkBox6 = (CheckBox) findViewById(R.id.checkBox6);

checkBox6.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {
            item6 = "Funciona";

        }
        else

```

```

        {
            item6 = "Falhou";
        }
    }
});

if (item6 == null)
    item6 = "Falhou";

CheckBox checkBox7 = (CheckBox) findViewById(R.id.checkBox7);

checkBox7.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item7 = "Funciona";

        }
        else
        {

            item7 = "Falhou";
        }

    }

});

if (item7 == null)
    item7 = "Falhou";

CheckBox checkBox8 = (CheckBox) findViewById(R.id.checkBox8);

checkBox8.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item8 = "Funciona";

        }
        else
        {

            item8 = "Falhou";
        }

    }

});

```

```

        if (item8 == null)
            item8 = "Falhou";

CheckBox checkBox9 = (CheckBox) findViewById(R.id.checkBox9);

checkBox9.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item9 = "Funciona";

        }
        else
        {

            item9 = "Falhou";
        }

    }

});

if (item9 == null)
    item9 = "Falhou";

CheckBox checkBox10 = (CheckBox) findViewById(R.id.checkBox10);

checkBox10.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item10 = "Funciona";

        }
        else
        {

            item10 = "Falhou";
        }

    }

});

if (item10 == null)
    item10 = "Falhou";

```

```

        Button Terminar = (Button) findViewById(R.id.terminar);
        Terminar.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {

                // TODO Auto-generated method stub
                mChronometer.stop();
                duracao = mChronometer.getText().toString();
                Estado_soc = GetEstado();
                equipamento = "VHF_DSC_Sailor";
                DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

                for(int i =0; i<Estado_soc.length; i++)
                    ActualizarEstado(Estado_soc[i]);

                ActualizarRegisto(duracao, equipamento);
                DisplayToast("Os dados da SOC foram guardados na BD");

            }

        });

    }

    public void ActualizarEstado (String EstadoEquipamento){

        String Resultado = "";

        try {

            urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/GravarVHF_Sailor.jsp?Estado="+EstadoEq
uipamento;

            Resultado = ConectarHTTP.executarHttpGet(urlGet);

            Log.i("conexão" , "resultado = " +Resultado);
            DisplayToast("Os dados da SOC foram guardados na BD");

        }

        catch (Exception erro){

            Log.i("erro" , "erro = " +erro);

            erro.printStackTrace();

        }

    }

```



```

    }

    public void AtualizarRegistro (String equipamento, String duracao ){

        String Resultado = "";

        try {

            urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/Registrar_tempo.jsp?Duracao="+duracao+"
&Equipamento="+equipamento;
            Resultado = ConectarHTTP.executarHttpGet(urlGet);

            Log.i("conexão" , "resultado = " +Resultado);

        }

        catch (Exception erro){

            Log.i("erro" , "erro = " +erro);
            erro.printStackTrace();

        }

    }

    public String[] GetEstado()
    {

        return estadoDsc= new String[] {item1, item2, item3, item4, item5,
item6, item7, item8, item9, item10};

    }

    // função que devolve a duração da SOC (em formato de texto)

    public String GetDuracao() {

        return duracao;

    }

    private void DisplayToast(String mensagem) {

        // TODO Auto-generated method stub
        Toast.makeText(getApplicationContext(), mensagem,
Toast.LENGTH_SHORT).show();

    }

```

```
}
```

### VHF\_Sealand:

```
package tese.sistema.socs;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Chronometer;
import android.widget.Toast;

public class VHF_Sealand extends Activity {

    Chronometer mChronometer;
    String urlGet;
    String item1,item2, item3, item4, item5, item6, item7, item8, item9,
    equipamento, duracao;
    String[] estadoSealand, Estado_soc;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        estadoSealand= new String[] {};
        super.onCreate(savedInstanceState);
        setContentView(R.layout.vhf_sealand);
        mChronometer = (Chronometer) findViewById(R.id.chronometer);
        mChronometer.start();

        StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        //Verificar Checkbox

        CheckBox checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
        checkBox1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub

                if(((CheckBox)v).isChecked())
                {

                    item1 = "Funciona";

                }
                else
                {
                    item1 ="Falhou";
                }
            }
        });
    }
}
```

```

    }
});

if (item1 == null)
    item1 = "Falhou";

CheckBox checkBox2 = (CheckBox) findViewById(R.id.checkBox2);
checkBox2.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item2 = "Funciona";

        }
        else
        {
            item2 = "Falhou";
        }
    }
});

if (item2 == null)
    item2 = "Falhou";

CheckBox checkBox3 = (CheckBox) findViewById(R.id.checkBox3);
checkBox3.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item3 = "Funciona";

        }
        else
        {
            item3 = "Falhou";
        }
    }
});

if (item3 == null)
    item3 = "Falhou";

```

```

CheckBox checkBox4 = (CheckBox) findViewById(R.id.checkBox4);

```

```

checkBox4.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item4 ="Funciona";

        }
        else
        {

            item4 ="Falhou";
        }

    }

});

if (item4 == null)
    item4 ="Falhou";

```

```

CheckBox checkBox5 = (CheckBox) findViewById(R.id.checkBox5);

```

```

checkBox5.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item5 ="Funciona";

        }
        else
        {

            item5 ="Falhou";
        }

    }

});

if (item5 == null)
    item5 ="Falhou";

```

```

CheckBox checkBox6 = (CheckBox) findViewById(R.id.checkBox6);

```

```

checkBox6.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item6 = "Funciona";

        }
        else
        {

            item6 = "Falhou";
        }

    }

});

if (item6 == null)
    item6 = "Falhou";

```

```

CheckBox checkBox7 = (CheckBox) findViewById(R.id.checkBox7);

```

```

checkBox7.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if(((CheckBox)v).isChecked())
        {

            item7 = "Funciona";

        }
        else
        {

            item7 = "Falhou";
        }

    }

});

if (item7 == null)
    item7 = "Falhou";

```

```

CheckBox checkBox8 = (CheckBox) findViewById(R.id.checkBox8);

```

```

checkBox8.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if((CheckBox)v.isChecked())
        {

            item8 = "Funciona";

        }
        else
        {

            item8 = "Falhou";
        }

    }

});

if (item8 == null)
    item8 = "Falhou";

CheckBox checkBox9 = (CheckBox) findViewById(R.id.checkBox9);

checkBox9.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        if((CheckBox)v.isChecked())
        {

            item9 = "Funciona";

        }
        else
        {

            item9 = "Falhou";
        }

    }

});

if (item9 == null)
    item9 = "Falhou";

Button Terminar = (Button) findViewById(R.id.terminar);
Terminar.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View v) {

    // TODO Auto-generated method stub
    mChronometer.stop();
    duracao = mChronometer.getText().toString();
    Estado_soc = GetEstado();
    equipamento = "VHF_Sealand";
    DisplayToast("Acabou de Efectuar a SOC do
Equipamento");

    for(int i =0; i<Estado_soc.length; i++)
        ActualizarEstado(Estado_soc[i]);

    ActualizarRegisto(duracao, equipamento);
    DisplayToast("Os dados da SOC foram guardados na BD");

}

});

}

public void ActualizarEstado (String EstadoEquipamento){

    String Resultado = "";

    try {

        urlGet = "http://
"+BuscarIP.getValue()+":8080/SocsWebbApp/GravarVHF_Sealand.jsp?Estado="+EstadoE
quipamento;

        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);
        DisplayToast("Os dados da SOC foram guardados na BD");

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        DisplayToast("Sem ligação a rede");
        erro.printStackTrace();

    }

}

public void ActualizarRegisto (String equipamento, String duracao ){

```

```

String Resultado = "";

    try {

        urlGet = "http://"
        "+BuscarIP.getValue()+":8080/SocsWebbApp/Registrar_tempo.jsp?Duracao="+duracao+"
        &Equipamento="+equipamento;
        Resultado = ConectarHTTP.executarHttpGet(urlGet);

        Log.i("conexão" , "resultado = " +Resultado);

    }

    catch (Exception erro){

        Log.i("erro" , "erro = " +erro);

        erro.printStackTrace();

    }

}

public String[] GetEstado()
{

    return estadoSealand= new String[] {item1, item2, item3, item4,
item5, item6, item7, item8};

}

private void DisplayToast(String mensagem) {

    Toast.makeText(getBaseContext(), mensagem,
    Toast.LENGTH_SHORT).show();

}

}

```

### ConectarHTTP:

```

package tese.sistema.socs;

import java.io.IOException;
import java.util.ArrayList;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URI;

import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;

```



```

import org.apache.http.conn.params.ConnManagerParams;
import org.apache.http.params.HttpConnectionParams;
import org.apache.http.params.HttpParams;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;

import android.util.Log;

public class ConectarHTTP {

    public static final int Http_Timeout = 30 * 7000;
    private static HttpClient httpClient;

    private static HttpClient getHttpClient(){
        if(httpClient == null){
            httpClient = new DefaultHttpClient();
            final HttpParams httpparams = httpClient.getParams();
            HttpConnectionParams.setConnectionTimeout(httpparams,
Http_Timeout);
            HttpConnectionParams.setSoTimeout(httpparams, Http_Timeout);
            ConnManagerParams.setTimeout(httpparams, Http_Timeout);

        }

        return httpClient;
    }

    public static String ejecutarHttpGet(String url) throws Exception{

        BufferedReader bufferedreader = null;

        try{
            HttpClient Client = getHttpClient();
            HttpGet httpGet = new HttpGet(url);

            httpGet.setURI(new URI(url));
            HttpResponse httpResponse = Client.execute(httpGet);
            bufferedreader =new BufferedReader(new
InputStreamReader(httpResponse.getEntity().getContent()));
            StringBuffer stringBuffer = new StringBuffer("");
            String line = "";
            String LS = System.getProperty("line.separator");

            while((line = bufferedreader.readLine())!=null){
                stringBuffer.append(line + LS);
            }
            bufferedreader.close();

            String resultado = stringBuffer.toString();
            return resultado;
        }
        finally {
            if(bufferedreader!=null){
                try{
                    bufferedreader.close();
                }
            }
        }
    }
}

```

```

    }
    catch (IOException e){
        Log.i("erro" , "ao conectar");
        e.printStackTrace();
    }
}
}
}
}

```

### BuscarIP:

```

package tese.sistema.socs;

public class BuscarIP {

    public static String valor_ip;

    public static String getValue() {
        return valor_ip;
    }

    public static void setValue(String valor_ip) {
        BuscarIP.valor_ip = valor_ip;
    }
}

```

### ManifestFile:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    package="tese.sistema.socs"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="17" android:maxSdkVersion="17"/>
    <uses-configuration />

    <uses-permission android:name="android.permission.INTERNET"/>

    <application

        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme"
        >

        <activity
            android:name="tese.sistema.socs.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>

```

```

        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

```

''' actividade para a Sereia

```

<activity
    android:label="Sereia"
    android:name=".Sereia"
    android:exported="false">
    <intent-filter >
        <action android:name="tese.sistema.Sereia"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>

</activity>

```

''' actividade para o Angulo do Leme

```

<activity
    android:label="Angulo_Leme"
    android:name=".Angulo_Leme"
    android:exported="false">
    <intent-filter >
        <action android:name="tese.sistema.Angulo_Leme"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>

</activity>

```

''' Actividade para o Piloto automático

```

<activity
    android:label="Piloto_Automatico"
    android:name=".Piloto_Automatico"
    android:exported="false">
    <intent-filter >
        <action android:name="tese.sistema.Piloto_Automatico"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>

</activity>

```

''' actividade para o projector de sinais de EB

```

<activity
    android:label="Projector_Sinais_EB"
    android:name=".Projector_Sinais_EB"
    android:exported="false">
    <intent-filter >
        <action android:name="tese.sistema.Projector_Sinais_EB"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>

</activity>

```

```

''' atividade para o projector de sinais de BB
<activity
    android:label="Projector_Sinais_BB"
    android:name=".Projector_Sinais_BB"
    android:exported="false">
        <intent-filter >
            <action android:name="tese.sistema.Projector_Sinais_BB"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

''' atividade para as adriças e balões
<activity
    android:label="Adrica_Baloes"
    android:name=".Adrica_Baloes"
    android:exported="false">
        <intent-filter >
            <action android:name="tese.sistema.Adrica_Baloes"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

''' atividade para Lanterna
<activity
    android:label="Lanterna"
    android:name=".Lanterna"
    android:exported="false">
        <intent-filter >
            <action android:name="tese.sistema.Lanterna"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

''' atividade para as Luzes Omnidirecionais
<activity
    android:label="Luzes_Omni"
    android:name=".Luzes_Omni"
    android:exported="false">
        <intent-filter >
            <action android:name="tese.sistema.Luzes_Omni"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

''' Conectar HTTP
<activity
    android:label="ConectarHTTP"
    android:name=".ConectarHTTP"
    android:exported="false">
        <intent-filter >
            <action android:name="tese.sistema.ConectarHTTP"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

```

```

''' OQP
<activity
    android:label="OQP"
    android:name=".OQP"
    android:exported="false">
        <intent-filter >
            <action android:name="tese.sistema.OQP"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

''' Marinheiro
<activity
    android:label="Marinheiro_Leme"
    android:name=".Marinheiro_Leme"
    android:exported="false">
        <intent-filter >
            <action android:name="tese.sistema.Marinheiro_Leme"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

''' Operador
<activity
    android:label="Operador_Sinais"
    android:name=".Operador_Sinais"
    android:exported="false">
        <intent-filter >
            <action android:name="tese.sistema.Operador_Sinais"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

''' KH
<activity
    android:label="KH_1007_NTD"
    android:name=".KH_1007_NTD"
    android:exported="false">
        <intent-filter >
            <action android:name="tese.sistema.KH_1007_NTD"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

''' Anemometro

    <activity
        android:label="Anemometro"
        android:name=".Anemometro"
        android:exported="false">

```

```

        <intent-filter >
            <action android:name="tese.sistema.Anemometro"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

    <activity
        android:label="DGPS"
        android:name=".DGPS"
        android:exported="false">
        <intent-filter >
            <action android:name="tese.sistema.DGPS"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

    <activity
        android:label="VHF_Sealand"
        android:name=".VHF_Sealand"
        android:exported="false">
        <intent-filter >
            <action android:name="tese.sistema.VHF_Sealand"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

    <activity
        android:label="VHF_DSC_Sailor"
        android:name=".VHF_DSC_Sailor"
        android:exported="false">
        <intent-filter >
            <action android:name="tese.sistema.VHF_DSC_Sailor"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

    <activity
        android:label="Sargento"
        android:name=".Sargento"
        android:exported="false">
        <intent-filter >
            <action android:name="tese.sistema.Sargento"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

```

```

<activity
    android:label="Luzes_Nav"
    android:name=".Luzes_Nav"
    android:exported="false">
    <intent-filter >
        <action android:name="tese.sistema.Luzes_Nav"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>

</activity>

</application>

</manifest>

```

## Anexo D – Código fonte dos *Web Services*

### GravarAdrica:

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>
Foram Gravadas Alterações na Base de Dados :<br/>

<%
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String guarda; // variavel que vai guardar as alterações na base de dados
String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+" "/" + "MM"+" "/" + "yy")).format(new
Date())).toString();
Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
guarda = "insert into adrica_baloes (Estado,Data) values" + "(' ' +
request.getParameter("Estado")+' ' , (''+Data+''))";

```

```
st.executeUpdate(guarda);
conn.close();
```

```
%>
```

```
</body>
```

```
</html>
```

## GravarAnemometro:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>
Foram Gravadas Alterações na Base de Dados :<br/> <br/>

<%
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+" "/"+"MM"+" "/"+"yy")).format(new
Date())).toString();

String guarda; // variavel que vai guardar as alterações na base de dados

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
guarda = "insert into anemometro (Estado,Data) values" + "(" +
request.getParameter("Estado")+"', ('"+Data+"'))";

st.executeUpdate(guarda);

conn.close();
```



```
%>

</body>

</html>
```

### GravarAngulo:

```
%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>
Foram Gravadas Alterações na Base de Dados :<br/> <br/>
```

```
<%
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();
String guarda; // variavel que vai guardar as alterações na base de dados

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
guarda = "insert into Angulos_Leme (Estado, Data) values" + "(' '+
request.getParameter("Estado")+",' (''+Data+'') )";

st.executeUpdate(guarda);

conn.close();
```

```
%>

</body>

</html>
```

## GravarBuzina:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>
A base de Dados foi Atualizada :<br/>
```

```
<%
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();
String guarda; // variavel que vai guardar as alterações na base de dados

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
guarda = "insert into sereia (Estado,Data) values" + "(" +
request.getParameter("Estado")+"', ('"+Data+"'))";

st.executeUpdate(guarda);

conn.close();

%>
```

```
</body>
```

```
</html>
```

### GravarDGPS:

```
%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>
Foram Gravadas Alterações na Base de Dados :<br/> <br/>
```

```
<%
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+" "/"+"MM"+" "/"+"yy")).format(new
Date())).toString();
String guarda; // variavel que vai guardar as alterações na base de dados

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);
st = conn.createStatement();

guarda = "insert into dgps (Estado, Data) values" + "(" +
request.getParameter("Estado")+"', ('"+Data+"') )";

st.executeUpdate(guarda);

conn.close();

%>

</body>

</html>
```

### GravarKH:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```

        pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>
Foram Gravadas Alterações na Base de Dados :<br/> <br/>

```

```

<%
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();
String guarda; // variavel que vai guardar as alterações na base de dados

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);
st = conn.createStatement();

guarda = "insert into kh (Estado, Data) values" + "(' " +
request.getParameter("Estado")+"', ('"+Data+"') )";

st.executeUpdate(guarda);

conn.close();

%>

</body>

</html>

```

### GravarLanterna:

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>
<html>

```

```

<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>
Foram Gravadas Alterações na Base de Dados :<br/> <br/>

<%
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+" "/"MM"+" "/"yy")).format(new
Date())).toString();
String guarda; // variavel que vai guardar as alterações na base de dados

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);
st = conn.createStatement();

guarda = "insert into lanterna (Estado, Data) values" + "(" + " " +
request.getParameter("Estado")+"', ("'+Data+'') )";

st.executeUpdate(guarda);

conn.close();

%>

</body>

</html>

```

## GravarLuzes:

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>

```

Foram Gravadas Alterações na Base de Dados :<br/> <br/>

```
<%
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+" "/"+"MM"+" "/"+"yy")).format(new
Date())).toString();
String guarda; // variavel que vai guardar as alterações na base de dados

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);
st = conn.createStatement();

guarda = "insert into luzes_omni (Estado, Data) values" + "(' "+
request.getParameter("Estado")+"', ('"+Data+"'))";

st.executeUpdate(guarda);

conn.close();

%>

</body>

</html>
```

### GravarLuzesNav:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>
Foram Gravadas Alterações na Base de Dados :<br/> <br/>
```

```
<%
```

```

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();
String guarda; // variavel que vai guardar as alterações na base de dados

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);
st = conn.createStatement();

guarda = "insert into luzes_Nav (Estado, Data) values" + "(' " +
request.getParameter("Estado")+"', ('"+Data+"') )";

st.executeUpdate(guarda);

conn.close();

%>

</body>

</html>

```

### GravarPiloto:

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>
Foram Gravadas Alterações na Base de Dados :<br/> <br/>

<%
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

```

```

String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();
String guarda; // variavel que vai guardar as alterações na base de dados

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);
st = conn.createStatement();

guarda = "insert into Piloto_Automatico (Estado, Data) values" + "(' "+
request.getParameter("Estado")+"', (' "+Data+"'))";

st.executeUpdate(guarda);

conn.close();

%>

</body>

</html>

```

## GravarProjectorBB:

```

    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>
Foram Gravadas Alterações na Base de Dados :<br/> <br/>

```

```

<%
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();
String guarda; // variavel que vai guardar as alterações na base de dados

Class.forName("com.mysql.jdbc.Driver").newInstance();

```



```

conn = DriverManager.getConnection(url);
st = conn.createStatement();

guarda = "insert into projector_bb (Estado, Data) values" + "(' " +
request.getParameter("Estado")+"', ('"+Data+"'))";

st.executeUpdate(guarda);

conn.close();

%>

</body>

</html>

```

### GravarVHF\_Sailor:

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>
Foram Gravadas Alterações na Base de Dados :<br/> <br/>

<%
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();
String guarda; // variavel que vai guardar as alterações na base de dados

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);
st = conn.createStatement();
guarda = "insert into vhf_sailor (Estado, Data) values" + "(' " +
request.getParameter("Estado")+"', ('"+Data+"'))";

```

```

st.executeUpdate(guarda);

conn.close();

%>

</body>

</html>

```

### GravarVHF\_Sealand:

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>
Foram Gravadas Alterações na Base de Dados :<br/> <br/>

```

```

<%
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();
String guarda; // variavel que vai guardar as alterações na base de dados

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);
st = conn.createStatement();
guarda = "insert into vhf_sailor (Estado,Data) values" + "(' "+
request.getParameter("Estado")+"', ('"+Data+"') )";

st.executeUpdate(guarda);

conn.close();

%>

</body>

```

```
</html>
```

## Registrar\_tempo:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import = "java.sql.Connection" %>
<%@ page import = "java.sql.DriverManager" %>
<%@ page import = "java.util.*" %>
<%@ page import = "java.text.*" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<body>
Foram Gravadas Alterações na Base de Dados :<br/> <br/>

<%
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String guarda_data;
String Data;
Data = (new Date()).toString();

Data = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();

String guarda; // variavel que vai guardar as alterações na base de dados

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
guarda = "insert into registo_tempo (Duracao, Equipamento, Data) values" +
"('"+request.getParameter("Equipamento")+"', ' "+
request.getParameter("Duracao")+"', ('"+Data+"'))";
st.executeUpdate(guarda);

conn.close();

%>

</body>

</html>
```

## Anexo E – Código fonte da Aplicação Web

### Index:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License

Name      : TwoColours
Description: A two-column, fixed-width design with dark color scheme.
Version    : 1.0
Released   : 20130811

-->
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title></title>
<meta name="keywords" content="" />
<meta name="description" content="" />
<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />

<script type="text/javascript">

<!--
function MM_jumpMenu(targ,selObj,restore){ //v3.0
    eval(targ+".location='"+selObj.options[selObj.selectedIndex].value+"'");
    if (restore) selObj.selectedIndex=0;
}
//-->
</script>

<!--[if IE 6]><link href="default_ie6.css" rel="stylesheet" type="text/css"
/><![endif]-->

</head>
<body>
<div id="header">
    <div id="menu" class="container">
        <ul>

            <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
            <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
            <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
```

```

        </ul>
    </div>
</div>
<div id="Logo" class="container">
    <h1><a href="#" class="icon icon-tasks"><span>SOCs</span></a></h1>
</div>
<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2>        SOCS Web Application        </h2>
        </div>

<div id="page" class="container">

<p align="left"><strong> Oficial de Quarto</strong>
    <select name="jumpMenu" id="jumpMenu"
onChange="MM_jumpMenu('parent',this,0)">
        <option value="Socs.jsp" selected>Equipamentos </option>
        <option value="KelvinHughes.jsp">KH 1007 NTD</option>
        <option value="DGPS_Furuno.jsp">DGPS FURUNO</option>
        <option value="Sealand.jsp">TX/RX VHF Sealand</option>
        <option value="Sailor.jsp">VHF DSC Sailor </option>
    </select>

    <p align="left"><strong>Sargento de Quarto </strong>
    <select name="jumpMenu2" id="jumpMenu2"
onChange="MM_jumpMenu('parent',this,0)">
        <option value="Socs.jsp" selected>Equipamentos </option>
        <option value="Anemometro.jsp">ANEMÔMETRO</option>
        <option value="Luzes_Nav.jsp">LUZES DE NAV.</option>
        <option value="sereia.jsp">SEREIA</option>
    </select>

    <p align="left"><strong>Marinheiro do Leme</strong>
    <select name="jumpMenu3" id="jumpMenu3"
onChange="MM_jumpMenu('parent',this,0)">
        <option value="Socs.jsp" selected>Equipamentos </option>
        <option value="Piloto_automatico.jsp">Piloto Automático</option>
        <option value="indicador_angulo.jsp">Indicador de Angulo do Leme</option>
    </select>

    <p align="left"><strong> Operador de Sinais</strong>
    <select name="jumpMenu4" id="jumpMenu4"
onChange="MM_jumpMenu('parent',this,0)">
        <option value="Socs.jsp" selected>Equipamentos </option>
        <option value="projector_EB.jsp">Projector de Sinais EB</option>
        <option value="Projector_BB.jsp">Projector de Sinais BB</option>
        <option value="Adrica.jsp">Adriça e Balões </option>
        <option value="Lanterna.jsp">Lanterna Portatil</option>
        <option value="Luzes_Omni.jsp">Luzes Omnidirecionais</option>
    </select>
</p>

</div>

        </div>
        <div id="sidebar"><a href="#" class="image image-full"></a></div>
</div>

```

```

        </div>
    </div>
</div>

<div id="copyright" class="container">
    <p>Desenvolvido por Adilson Euclides Maquita António | Tese de Mestrado
2013 </a> | Otimização das Standard Operator Checks (SOCs) - Utilização de
Dispositivos móveis na execução das SOC's </a>.</p>
</div>
</body>
</html>

```

## Adrica:

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License

Name      : TwoColours
Description: A two-column, fixed-width design with dark color scheme.
Version   : 1.0
Released  : 20130811

-->
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.*" %>
<%@ page import = "java.util.*" %>

<html>
<head>

<meta name="keywords" content="" />
<meta name="description" content="" />
<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
<title> </title>

</head>

<body>

<div id="header">
    <div id="menu" class="container">
        <ul>

            <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
            <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>

```

```

        <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
    </ul>
</div>
</div>

<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2>    Adriças e Balões    </h2>
        </div>
    </div>
</div>

<%

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs, rs2;
String xt,xt2;
String DataAtual;
String estadoAtual="", duracaoSOC="", equipamento;
String estadoArray[];

estadoArray = new String[7];
int i = 0;
int j =0;

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
DataAtual = (new Date()).toString();

DataAtual = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();

xt = "select * from adrica;";
xt2 = "select * from registo_tempo; ";

rs2 = st.executeQuery(xt2);

while (rs2.next())
{
    String DataBD2;
    DataBD2 = rs2.getString("Data");
    equipamento = rs2.getString("Equipamento");

    if(DataAtual.equals(DataBD2) && equipamento.equals("Adrica)){
        duracaoSOC = rs2.getString("Duracao");
    }
}

```

```

    }

rs = st.executeQuery(xt);

while (rs.next())
{
    String DataBD;
    DataBD = rs.getString("Data");

    if(DataActual.equals(DataBD)){
        estadoActual = rs.getString("Estado");
        estadoArray[i]=estadoActual;
        i++;
    }

}

}

%>

<table width="662" border="1">
    <tr>
        <td width="525"><div align="justify">Itens da SOC </div></td>
        <td width="121"><div align="justify">Estado </div></td>
    </tr>
    <tr>
        <td><p><strong>1.</strong> No parque de sinais, todas as adriças estão em bom estado?</p></td>
        <td><%=estadoArray[0]%></td>
    </tr>
    <tr>
        <td><p><strong>2.</strong> Todas as adriças correm bem nas respectivas roldanas?</p></td>
        <td><%=estadoArray[1]%></td>
    </tr>
    <tr>
        <td><p><strong>3.</strong> Os gatos de cada adriça não apresentam sinais de desgaste?</p></td>
        <td><%=estadoArray[2]%></td>
    </tr>
    <tr>
        <td><br>
        <p><strong>4.</strong> Os balões de sinais estão disponíveis e devidamente peados?</p></td>
        <td><%=estadoArray[3]%></td>
    </tr>
    <tr>
        <td><br>
        <p><strong>5.</strong> A Bandeira Nacional de dois panos para usar com o navio no mar está em bom estado?<br>
        <br>
        </p></td>
        <td><%=estadoArray[4]%></td>
    </tr>
    <tr>

```



```

        <td><br>
        <strong>6.</strong> A flâmula está em bom estado?</td>
        <td><%=estadoArray[5]%></td>
    </tr>
    <tr>
        <td><strong>7.</strong> O Jaque pronto a usar está em bom estado?</td>
        <td><%=estadoArray[6]%></td>

    </tr>
</table>
Tempo padrão : 2:30 min
<p>Data: <%=DataActual %> </p>
<p>&nbsp;</p>

</body>

</html>

```

## Anemometro:

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License

Name      : TwoColours
Description: A two-column, fixed-width design with dark color scheme.
Version   : 1.0
Released  : 20130811

-->
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.*" %>
<%@ page import = "java.util.*" %>

<html>
<head>

<meta name="keywords" content="" />
<meta name="description" content="" />
<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
<title> </title>

</head>

<body>

<div id="header">

```

```

        <div id="menu" class="container">
            <ul>

                <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
                <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
                <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
            </ul>
        </div>
    </div>

    <div id="page" class="container">
        <div id="content">
            <div class="title">
                <h2> Anemómetro </h2>
            </div>
        </div>
    </div>

<%

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs, rs2;
String xt,xt2;
String DataAtual;
String estadoAtual="", duracaoSOC="", equipamento;
String estadoArray[];

estadoArray = new String[100];
int i = 0;
int j =0;

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
DataAtual = (new Date()).toString();

DataAtual = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();

xt = "select * from anemometro; ";
xt2 = "select * from registro_tempo; ";

rs2 = st.executeQuery(xt2);

while (rs2.next())
{
    String DataBD2;
    DataBD2 = rs2.getString("Data");
    equipamento = rs2.getString("Equipamento");

```

```

        if(DataActual.equals(DataBD2) && equipamento.equals("Anemometro")){
            duracaoSOC = rs2.getString("Duracao");
        }

    }

    rs = st.executeQuery(xt);

    while (rs.next())
    {
        String DataBD;
        DataBD = rs.getString("Data");

        if(DataActual.equals(DataBD)){
            estadoActual = rs.getString("Estado");
            estadoArray[i]=estadoActual;
            i++;
        }

    }

}

%>
<table width="754" border="1">
    <tr>
        <td width="653">Itens da SOC </td>
        <td width="85">Estado</td>
    </tr>
    <tr>
        <td>
            <u>Nos indicadores existentes na consola por cima da mesa das cartas, proceda à seguinte verificação:<br />
            <strong>(1)</strong> Os indicadores de direcção e velocidade do vento apresentam variação?</td>
        <td><%=estadoArray[0]%></td>
    </tr>
    <tr>
        <td><br />
            <strong>(2)</strong> Com velocidades de vento inferiores a 25 nós, premindo o botão branco, o ponteiro deflecte para a direita, apresentando uma indicação correcta na escala interior?</td>
        <td><%=estadoArray[1]%></td>
    </tr>
    <tr>
        <td><br />
            <strong>(3)</strong> A direcção e velocidade do vento registada são idênticas à indicada nos indicadores existentes a vante da Ponte?</td>
        <td><%=estadoArray[2]%></td>
    </tr>
    <tr>
        <td><br />
            <strong>(4)</strong> Actuando o reóstato de regulação de brilho, a iluminação dos indicadores varia? </td>
        <td><%=estadoArray[3]%></td>
    </tr>
</table>

```

```
<p>Tempo padrão : 1:00 min </p>
<p>Data: <%=DataActual %> </p>
```

```
<p>&nbsp;</p>
```

```
</body>
</html>
```

## DGPS\_Furuno:

```
<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
```

Design by Free CSS Templates

<http://www.freecsstemplates.org>

Released for free under a Creative Commons Attribution 2.5 License

Name : TwoColours

Description: A two-column, fixed-width design with dark color scheme.

Version : 1.0

Released : 20130811

```
-->
```

```
<%@ page import = "java.util.Date" %>
```

```
<%@ page import = "java.text.*" %>
```

```
<%@ page import = "java.util.*" %>
```

```
<html>
```

```
<head>
```

```
<meta name="keywords" content="" />
```

```
<meta name="description" content="" />
```

```
<link
```

```
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />
```

```
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
```

```
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
```

```
<title> </title>
```

```
</head>
```

```
<body>
```

```
<div id="header">
```

```
<div id="menu" class="container">
```

```
<ul>
```

```
<li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
```

```
<li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
```

```
<li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
```

```
</ul>
```

```
</div>
```

```

</div>

<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2>    DGSP    </h2>
        </div>
    </div>
</div>
<table width="882" border="1">

<%

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs, rs2;
String xt,xt2;
String DataAtual;
String estadoAtual="", duracaoSOC="", equipamento;
String estadoArray[];

estadoArray = new String[100];
int i = 0;
int j =0;

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
DataAtual = (new Date()).toString();

DataAtual = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();

xt = "select * from dgps; ";
xt2 = "select * from registo_tempo; ";
//rs = st.executeQuery(xt);
rs2 = st.executeQuery(xt2);

while (rs2.next())
{
    String DataBD2;
    DataBD2 = rs2.getString("Data");
    equipamento = rs2.getString("Equipamento");

    if(DataAtual.equals(DataBD2) && equipamento.equals("DGPS")){
        duracaoSOC = rs2.getString("Duracao");
        out.println(rs2.getString("Duracao"));
    }
}

```

```

    }

rs = st.executeQuery(xt);
while (rs.next())
{
    String DataBD;
    DataBD = rs.getString("Data");

    if(DataActual.equals(DataBD)){
        estadoActual = rs.getString("Estado");
        estadoArray[i]=estadoActual;
        i++;
    }

}

}

%>
<tr>
<td width="728">Itens da SOC</td>
<td width="138">Estado </td>
</tr>
<tr>
<td><strong>(1)</strong> Estabeleça o equipamento, pressionando o botão
POWER.</td>
<td><%=estadoArray[0]%></td>
</tr>
<tr>
<td><strong>(a) </strong>O ecrã liga?</td>
<td><%=estadoArray[1]%></td>
</tr>
<tr>
<td><strong>(2)</strong> Premir tecla <strong>DISPLAY</strong>, seleccionar
<strong>DATA, premir </strong> <strong>ENTER</strong>. Aparece info. de pst
geográfica?</td>
<td><%=estadoArray[2]%></td>
</tr>
<tr>
<td><strong>(3) </strong>A indicação <strong>DGPS 3D </strong>aparece no
canto superior direito? (só aplicável em áreas com cobertura DGPS)</td>
<td><%=estadoArray[3]%></td>
</tr>
<tr>
<td><strong>(4) Premir tecla MENU|</strong><strong>ESC, </strong>aceder
opção de menu <strong>8. </strong><strong>Self Test</strong>s, aceder opção
<strong>4. Automatic Testing</strong>. Premir <strong>ENT</strong>.</td>
<td><%=estadoArray[4]%></td>
</tr>
<tr>
<td>Todos os resultados OK? </td>
<td><%=estadoArray[5]%></td>
</tr>
</table>
<p>Duração da SOC: <%=
rs2 = st.executeQuery(xt2);
while (rs2.next())

```

```

{
    String DataBD2;
    DataBD2 = rs2.getString("Data");
    equipamento = rs2.getString("Equipamento");

    if(DataActual.equals(DataBD2) && equipamento.equals("DGPS")){
        duracaoSOC = rs2.getString("Duracao");
        out.println(duracaoSOC);
        out.print(DataBD2);
        out.print(equipamento);
    }

} %></p>
<p>Tempo padrão : 2:00 min</p>
<p>Data: <%=DataActual %> </p>
</body>
</html>

```

## Indicador\_angulo:

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License

Name      : TwoColours
Description: A two-column, fixed-width design with dark color scheme.
Version   : 1.0
Released  : 20130811

-->
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.*" %>
<%@ page import = "java.util.*" %>

<html>
<head>

<meta name="keywords" content="" />
<meta name="description" content="" />
<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,70
0,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
<title> </title>

</head>

<body>

```

```

<div id="header">
    <div id="menu" class="container">
        <ul>

            <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
            <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
            <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
        </ul>
    </div>
</div>

<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2> Indicador de Ângulo do Leme </h2>
        </div>
    </div>
</div>

<%

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;
String xt,xt2;
String DataAtual;
String estadoAtual="", duracaoSOC="", equipamento;
String estadoArray[];

estadoArray = new String[11];
int i = 0;
int j =0;

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
DataAtual = (new Date()).toString();

DataAtual = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();

xt = "select * from angulos_leme; ";
xt2 = "select * from registo_tempo; ";
rs = st.executeQuery(xt);
rs = st.executeQuery(xt2);

while (rs.next())
{
    String DataBD2;

```



```

DataBD2 = rs.getString("Data");
equipamento = rs.getString("Equipamento");

if(DataActual.equals(DataBD2) && equipamento.equals("Angulo_Leme")){
    duracaoSOC = rs.getString("Duracao");
}

}

while (rs.next())
{
    String DataBD;
    DataBD = rs.getString("Data");

    if(DataActual.equals(DataBD)){
        estadoActual = rs.getString("Estado");
        estadoArray[i]=estadoActual;
        i++;
    }

}

%>

<table width="919" border="1">
    <tr>
        <td width="832">Itens da SOC</td>
        <td width="95">Estado</td>
    </tr>
    <tr>
        <td><p><strong>Acções Preliminares:</strong> <br>
- Preposicionar os Vigias nas asas da Ponte e o Cabo de Quarto na Casa da
Máquina do Leme.</p></td>
        <td><%=estadoArray[0]%></td>
    </tr>
    <tr>
        <td><p>- Estabeleça comunicações através da conferência apropriada com a
Casa da Máquina do Leme, MCR e Centro de Operações.</p></td>
        <td><%=estadoArray[1]%></td>
    </tr>
    <tr>
        <td><p>- Peça autorização ao Oficial de Quarto à Ponte para verificar os
indicadores de ângulo de leme.</p></td>
        <td><%=estadoArray[2]%></td>
    </tr>
    <tr>
        <td><p><strong>Verificação</strong> <strong>a efectuar:</strong> <br>
<strong>(1)</strong> Introduza os seguintes ângulos de leme e informe
as diferentes estações:</p>
        <td><%=estadoArray[3]%></td>
    </tr>
    <tr>
        <td><p>

```

```

10º de leme a EB</td>
<td><%=estadoArray[4]%></td>
</tr>
<tr>
<td><p>
10º de leme a BB.
</p></td>
<td><%=estadoArray[5]%></td>
</tr>
<tr>
<td><p><strong>(2) </strong>Verifique que nos diferentes indicadores, as
leituras são correctas:</p>
<p>Marinheiro do Leme;</p></td>
<td><%=estadoArray[6]%></td>
</tr>
<tr>
<td><p>Asa da Ponte de EB;
</p></td>
<td><%=estadoArray[7]%></td>
</tr>
<tr>
<td><p>C0;
</p></td>
<td>&nbsp;</td>
</tr>
<tr>
<td>>MCR;</td>
<td><%=estadoArray[8]%></td>
</tr>
<tr>
<td>Indicador da Máquina do Leme;</td>
<td><%=estadoArray[9]%></td>
</tr>
<tr>
<td>< height="62">Axiómetro do Aparelho do Leme
<p>&nbsp;</p></td>
<td><%=estadoArray[10]%></td>
</tr>
</table>
<p>Tempo padrão : 5:00 min</p>
<p>Data: <%=DataActual %> </p>

```

```

<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
</body>
</html>

```

## KelvinHughes:

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates

```

http://www.freecsstemplates.org  
Released for free under a Creative Commons Attribution 2.5 License

Name : TwoColours  
Description: A two-column, fixed-width design with dark color scheme.  
Version : 1.0  
Released : 20130811

```
-->
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.*" %>
<%@ page import = "java.util.*" %>

<html>
<head>

<meta name="keywords" content="" />
<meta name="description" content="" />
<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
<title> </title>

</head>

<body>

<div id="header">
    <div id="menu" class="container">
        <ul>

            <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
            <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
            <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
        </ul>
    </div>
</div>

<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2> Kelvin Hughes </h2>
        </div>
    </div>
</div>

<%

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs,rs2;
String xt,xt2;
String DataActual, DataBD2;
```

```

String estadoActual="", duracaoSOC="", equipamento;
String estadoArray[];

estadoArray = new String[100];
int i = 0;
int j =0;

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
DataActual = (new Date()).toString();

DataActual = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();

xt = "select * from kh; ";
xt2 = "select * from registo_tempo; ";

rs2 = st.executeQuery(xt2);

while (rs2.next())
{

    DataBD2 = rs2.getString("Data");
    equipamento = rs2.getString("Equipamento");

    if(DataActual.equals(DataBD2) && equipamento.equals("KH_1007_NTD")){
        duracaoSOC = rs2.getString("Duracao");

    }

}

rs = st.executeQuery(xt);
while (rs.next())
{
    String DataBD;
    DataBD = rs.getString("Data");

    if(DataActual.equals(DataBD)){
        estadoActual = rs.getString("Estado");
        estadoArray[i]=estadoActual;
        i++;

    }

}

%>
<table width="891" border="1">
<tr>

```

<div align="left"> <p>Itens da SOC</p> <p>Estado</p> </div>
---

Para a total execução deste SOC, a consola a testar deve ser MASTER.

(Se necessário premir "REQUEST MASTER")

**(1)** No Display deverá assegurar as seguintes condições de modo a obter um bom desempenho no SOC:

a) GANH0 entre 70 e 80;	<%=estadoArray[0] %>
b) SEA no mínimo nunca zero;	<%=estadoArray[1] %>
c) RAIN no mínimo.	<%=estadoArray[2] %>

**(3)** Seleccionar a escala 12NM. No menu SELECT RADAR seleccionar TX MON VIDEO ON

Aparece um "Sol" no centro do Radar cujo valor típico é 1NM e um "Pluma" com um comprimento típico de 6NM.

<%=estadoArray[3] %>

**(4)** Verifique que o Valor do HEADING lido no display (canto superior direito) é igual ao Repetidora da Giro.

<%=estadoArray[4] %>

**(5)** Verifique que o valor de STW (velocidade) no display (canto superior direito) é igual ao da repetidora de Odómetro.

<%=estadoArray[5] %>

**(6)** Compare a posição GPS indicada no canto inferior direito, com a posição do respectivo GPS seleccionado.

<%=estadoArray[6] %>

## Lanterna:

```
<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License
```

Name : TwoColours  
Description: A two-column, fixed-width design with dark color scheme.  
Version : 1.0  
Released : 20130811

```
-->
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.*" %>
<%@ page import = "java.util.*" %>

<html>
<head>

<meta name="keywords" content="" />
<meta name="description" content="" />
<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
<title> </title>

</head>

<body>

<div id="header">
    <div id="menu" class="container">
        <ul>
            <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
            <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
            <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
        </ul>
    </div>
</div>

<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2> Lanterna de sinal Portátil </h2>
        </div>
    </div>
</div>

<%

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;
String xt,xt2;
String DataAtual;
String estadoAtual="", duracaoSOC="", equipamento;
String estadoArray[];
```

```

estadoArray = new String[4];
int i = 0;
int j = 0;

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
DataActual = (new Date()).toString();

DataActual = ((new SimpleDateFormat("dd"+" "/" + "MM"+" "/" + "yy")).format(new
Date())).toString();

xt = "select * from lanterna; ";
xt2 = "select * from registo_tempo; ";
rs = st.executeQuery(xt);
rs = st.executeQuery(xt2);

while (rs.next())
{
    String DataBD2;
    DataBD2 = rs.getString("Data");
    equipamento = rs.getString("Equipamento");

    if(DataActual.equals(DataBD2) && equipamento.equals("Lanterna")){
        duracaoSOC = rs.getString("Duracao");
    }

}

while (rs.next())
{
    String DataBD;
    DataBD = rs.getString("Data");

    if(DataActual.equals(DataBD)){
        estadoActual = rs.getString("Estado");
        estadoArray[i]=estadoActual;
        i++;
    }

}

%>
<table width="687" border="1">
<tr>
<td width="533"><div align="justify">Itens da SOC </div></td>
<td width="138"><div align="justify">Estado </div></td>
</tr>

```

```

<tr>
  <td><p>Retire a lanterna da caixa respectiva e ligue o cabo de alimentação
na ficha respectiva.</p>
  <p><strong>1.</strong> Verifique que o vidro e o espelho da óptica
estão em boas condições?</p></td>
  <td><%=estadoArray[0]%></td>
</tr>
<tr>
  <td><p><strong>2.</strong> Coloque o comutador ON/OFF na posição ON.
Verifique que a lâmpada se acende?</p></td>
  <td><%=estadoArray[1]%>;</td>
</tr>
<tr>
  <td><p><strong>3.</strong> Destrave o gatilho da lanterna. Accionando o
gatilho o ocultador da lâmpada movimenta-se sem problemas?</p></td>
  <td><%=estadoArray[2]%></td>
</tr>
<tr>
  <td>Volte a colocar a lanterna na caixa.
  <p></p></td><td><%=estadoArray[3]%></td>
</tr>
</table>
<p>Tempo padrão : 1:00 min </p>
<p>Data: <%=DataActual %> </p>
</body>
</html>

```

## Luzes\_Nav:

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License

Name      : TwoColours
Description: A two-column, fixed-width design with dark color scheme.
Version   : 1.0
Released  : 20130811

-->
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.*" %>
<%@ page import = "java.util.*" %>

<html>
<head>

<meta name="keywords" content="" />
<meta name="description" content="" />
<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,70
0,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />

```



```

<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
<title> </title>

</head>

<body>

<div id="header">
    <div id="menu" class="container">
        <ul>

            <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
            <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
            <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
        </ul>
    </div>
</div>

<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2> Luzes de Navegação </h2>
        </div>
    </div>
</div>
<%

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;
String xt,xt2;
String DataActual;
String estadoActual="", duracaoSOC="", equipamento;
String estadoArray[];

estadoArray = new String[6];
int i = 0;
int j =0;

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
DataActual = (new Date()).toString();

DataActual = ((new SimpleDateFormat("dd"+" "/" + "MM"+" "/" + "yy")).format(new
Date())).toString();

xt = "select * from luzes_nav; ";
xt2 = "select * from registo_tempo; ";
rs = st.executeQuery(xt);
rs = st.executeQuery(xt2);

```

```

while (rs.next())
{
    String DataBD2;
    DataBD2 = rs.getString("Data");
    equipamento = rs.getString("Equipamento");

    if(DataActual.equals(DataBD2) && equipamento.equals("Luzes")){
        duracaoSOC = rs.getString("Duracao");
    }

}

while (rs.next())
{
    String DataBD;
    DataBD = rs.getString("Data");

    if(DataActual.equals(DataBD)){
        estadoActual = rs.getString("Estado");
        estadoArray[i]=estadoActual;
        i++;
    }

}

%>
<table width="693" border="1">
    <tr>
        <td width="536">Itens da SOC </td>
        <td width="141">Estado</td>
    </tr>
    <tr>
        <td><strong>1)</strong> Colocar o selector FEEDING na posição I. Os 3
        Led's verdes ON, Alim. I e Alim. II acendem? </td>
        <td><%=estadoArray[0]%></td>
    </tr>
    <tr>
        <td><strong>(2)</strong> Colocar o selector FEEDING na posição II. Os 3
        Led's verdes ON, Alim. I e Alim. II acendem? </td>
        <td><%=estadoArray[1]%></td>
    </tr>
    <tr>
        <td><strong>(3)</strong> Com o selector FEEDING na posição I ou II:<br />
        <strong>(a) </strong>Os indicadores Led's e o besouro actuam ao premir o
        botão REST LED?</td>
        <td><%=estadoArray[2]%></td>
    </tr>
    <tr>
        <td><strong>(b) </strong>Todos os Led's de alimentação (vermelhos) e de
        posição acendem ap premir o botão correspondente?</td>
        <td><%=estadoArray[3]%></td>
    </tr>

```

```

        <tr>
            <td><br />
                <strong>(4) </strong>Ao rodar o reóstato DIMMER, o brilho dos Led's
                aumenta e diminui? </td>
            <td><%=estadoArray[4]%></td>
        </tr>
        <tr>
            <td><strong>(5) </strong>Ao comutar o selector de brilho, a intensidade
            das lâmpadas dos faróis/luzes de sinalização varia? </td>
            <td><%=estadoArray[5]%></td>
        </tr>
    </table>
    <p>Tempo padrão :</p>
    <p>Data: <%=DataActual %> </p>

</body>
</html>

```

## Luzes\_Omni:

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License

Name      : TwoColours
Description: A two-column, fixed-width design with dark color scheme.
Version    : 1.0
Released   : 20130811

-->
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.*" %>
<%@ page import = "java.util.*" %>

<html>
<head>

<meta name="keywords" content="" />
<meta name="description" content="" />
<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,70
0,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
<title> </title>

</head>

<body>

<div id="header">
    <div id="menu" class="container">
        <ul>

```

```

        <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
        <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
        <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
    </ul>
</div>
</div>

<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2> Luzes Omnidirecionais </h2>
        </div>
    </div>
</div>
<%

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;
String xt,xt2;
String DataAtual;
String estadoAtual="", duracaoSOC="", equipamento;
String estadoArray[];

estadoArray = new String[8];
int i = 0;
int j =0;

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
DataAtual = (new Date()).toString();

DataAtual = ((new SimpleDateFormat("dd"/+"MM"/+"yy")).format(new
Date())).toString();

xt = "select * from luzes_omni; ";
xt2 = "select * from registo_tempo; ";
rs = st.executeQuery(xt);
rs = st.executeQuery(xt2);

while (rs.next())
{
    String DataBD2;
    DataBD2 = rs.getString("Data");
    equipamento = rs.getString("Equipamento");

    if(DataAtual.equals(DataBD2) && equipamento.equals("Luzes_Omni")){
        duracaoSOC = rs.getString("Duracao");
    }
}

```

```

    }

    }

while (rs.next())
{
    String DataBD;
    DataBD = rs.getString("Data");

    if(DataActual.equals(DataBD)){
        estadoActual = rs.getString("Estado");
        estadoArray[i]=estadoActual;
        i++;
    }

}

}

%>
<table width="891" border="1">
    <tr>
        <td width="180"><div align="left"> Itens da SOC</div></td>
        <td width="8">Estado </td>
    </tr>
    <tr>
        <td><p>No Pannel de Alarmes e Segurança da Navegação (Ponte) coloque o selector de alimentação &quot;I/II&quot; para a posição &quot;I&quot;</p></td>
        <td><%=estadoArray[0]%></td>
    </tr>
    <tr>
        <td><p>&nbsp;</p>
        <p><strong>1.</strong> Accionando o interruptor correspondente a &quot;POSIÇÃO/SINAIS MORSE&quot; os dois LEDs amarelos acendem não soando qualquer alarme sonoro?</p></td>
        <td><%=estadoArray[1]%></td>
    </tr>
    <tr>
        <td><strong>2.</strong> Verifique no Mastro que as luzes se encontram acesas?</td>
        <td><%=estadoArray[2]%></td>
    </tr>
    <tr>
        <td>Desligue o interruptor correspondente a &quot;POSIÇÃO/SINAIS MORSE&quot;.</td>
        <td><%=estadoArray[3]%></td>
    </tr>
    <tr>
        <td><strong>3.</strong> Verifique com a colaboração dos Vigias que accionando as seguintes Chaves Morse as lâmpadas acendem:<br>
        <strong>a.</strong> Asa da Ponte de EB?</td>
        <td><%=estadoArray[4]%></td>
    </tr>
    <tr>
        <td><strong>b.</strong> Asa da Ponte de BB?</td>
        <td><%=estadoArray[5]%></td>
    </tr>

```

```

        <td><%=estadoArray[5]%></td>
    </tr>
    <tr>
        <td><strong>c.</strong> Ponte (junto do Sealand 66)?</td>
        <td><%=estadoArray[6]%></td>
    </tr>
    <tr>
        <td>Desligue o comutador de alimentação do Paine! de Alarmes e Segurança
da Navegação, colocando-o na posição 0.</td>
        <td><%=estadoArray[7]%></td>
    </tr>
</table>
<p>Tempo padrão: 1:30</p>
<p>Data: <%=DataActual %> </p>

</body>
</html>

```

### Piloto\_automatico:

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License

Name      : TwoColours
Description: A two-column, fixed-width design with dark color scheme.
Version   : 1.0
Released  : 20130811

-->
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.*" %>
<%@ page import = "java.util.*" %>

<html>
<head>

<meta name="keywords" content="" />
<meta name="description" content="" />
<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,70
0,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
<title> </title>

</head>

<body>

<div id="header">
    <div id="menu" class="container">
        <ul>

```

```

        <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
        <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
        <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
    </ul>
</div>
</div>

<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2> Piloto Automático </h2>
        </div>
    </div>
</div>
<%

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;
String xt,xt2;
String DataAtual;
String estadoAtual="", duracaoSOC="", equipamento;
String estadoArray[];

estadoArray = new String[10];
int i = 0;
int j =0;

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
DataAtual = (new Date()).toString();

DataAtual = ((new SimpleDateFormat("dd"/"MM"/"yy")).format(new
Date())).toString();

xt = "select * from piloto_automatico; ";
xt2 = "select * from registo_tempo; ";
rs = st.executeQuery(xt);
rs = st.executeQuery(xt2);

while (rs.next())
{
    String DataBD2;
    DataBD2 = rs.getString("Data");
    equipamento = rs.getString("Equipamento");

    if(DataAtual.equals(DataBD2) &&
equipamento.equals("Piloto_Automatico")){

```

```

        duracaoSOC = rs.getString("Duracao");

    }

}

while (rs.next())
{
    String DataBD;
    DataBD = rs.getString("Data");

    if(DataActual.equals(DataBD)){
        estadoActual = rs.getString("Estado");
        estadoArray[i]=estadoActual;
        i++;
    }

}

}

%>
<table width="895" border="1">
    <tr>
        <td width="742">Itens da SOC</td>
        <td width="137">Estado Actual </td>
    </tr>
    <tr>
        <td><p>Sem ter o Piloto Automático seleccionado (i. é, com o governo do navio a ser feito pela roda do leme ou pelo manípulo) proceda às seguintes verificações:<br>
                <strong>(1)</strong> Existe iluminação no display?</p></td>
        <td><%=estadoArray[0]%></td>
    </tr>
    <tr>
        <td><strong>2)</strong> Premindo a tecla TEST RESET do painel superior NAUTOPILOT D, todos os indicadores iluminam-se?</td>
        <td><%=estadoArray[1]%></td>
    </tr>
    <tr>
        <td><p><strong>(3)</strong> Rodando o botão rotativo central, o brilho do display varia?</p></td>
        <td><%=estadoArray[2]%></td>
    </tr>
    <tr>
        <td><p><strong>(4)</strong> Premindo as teclas de cursor (seta apontada para cima e seta apontada para baixo), os indicadores que estão por cima destas teclas iluminam-se sequencialmente a cada toque, e no display com a legenda SET COURSE, aparece um valor dentro dos limites indicados para cada indicador?</p></td>
        <td><%=estadoArray[3]%></td>
    </tr>
    <tr>
        <td><p><strong>(5)</strong> Premindo a tecla TEST RESET do painel inferior NAUTOALARM, verifique se:<br>
                <strong>(a)</strong> Todos os indicadores acendem.</p></td>
        <td><%=estadoArray[4]%></td>
    </tr>

```



```

</tr>
<tr>
  <td><p><strong>(b)</strong> O alarme sonoro soa?</p></td>
  <td><%=estadoArray[5]%></td>
</tr>
<tr>
  <td><p><strong>(6)</strong> A proa indicada no display HEADING coincide com
a proa do navio lida nas repetidoras da giro?</p></td>
  <td><%=estadoArray[6]%></td>
</tr>
<tr>
  <td><p><strong>(7)</strong> Ao comutar o seleccionador AUTO PILOT/RODA DE
LEME/MANÍPULO no painel inferior, o indicador respectivo acende-se quando é
seleccionada a posição de:<br>
<strong>(a)</strong> AUTO PILOT</p></td>
  <td><%=estadoArray[7]%></td>
</tr>
<tr>
  <td><p><strong>(b)</strong> RODA DE LEME</p></td>
  <td><%=estadoArray[8]%></td>
</tr>
<tr>
  <td><strong>(c)</strong> MANIPULO</td>
  <td><%=estadoArray[9]%></td>
</tr>
</table>
<p>Tempo padrão 1:30</p>
<p>Data: <%=DataActual %> </p>
</body>
</html>

```

## Projector\_BB:

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License

Name      : TwoColours
Description: A two-column, fixed-width design with dark color scheme.
Version   : 1.0
Released  : 20130811

-->
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.*" %>
<%@ page import = "java.util.*" %>

<html>
<head>

<meta name="keywords" content="" />
<meta name="description" content="" />

```

```

<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,70
0,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
<title> </title>

</head>

<body>

<div id="header">
    <div id="menu" class="container">
        <ul>

            <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
            <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
            <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
        </ul>
    </div>
</div>

<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2>    Projetor de BB    </h2>
        </div>
    </div>
</div>
<%

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;
String xt,xt2;
String DataAtual;
String estadoAtual="", duracaoSOC="", equipamento;
String estadoArray[];

estadoArray = new String[6];
int i = 0;
int j =0;

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
DataAtual = (new Date()).toString();

DataAtual = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();

xt = "select * from projector_bb; ";

```

```

xt2 = "select * from registo_tempo; ";
rs = st.executeQuery(xt);
rs = st.executeQuery(xt2);

while (rs.next())
{
    String DataBD2;
    DataBD2 = rs.getString("Data");
    equipamento = rs.getString("Equipamento");

    if(DataActual.equals(DataBD2) &&
    equipamento.equals("Projector_Sinais_BB")){
        duracaoSOC = rs.getString("Duracao");

    }

}

while (rs.next())
{
    String DataBD;
    DataBD = rs.getString("Data");

    if(DataActual.equals(DataBD)){
        estadoActual = rs.getString("Estado");
        estadoArray[i]=estadoActual;
        i++;

    }

}

%>
<table width="797" border="1">
    <tr>
        <td width="676">Itens da SOC</td>
        <td width="105">Estado da SOC</td>
    </tr>
    <tr>
        <td><p><strong>1.</strong> Desaperte os fixadores e verifique o movimento do projector no plano vertical e no plano horizontal. Ele roda sem problemas?</p></td>
        <td><%=estadoArray[0]%></td>
    </tr>
    <tr>
        <td><p><strong>2.</strong> Accionando a alavanca a persiana abre bem?</p></td>
        <td><%=estadoArray[1]%></td>
    </tr>
    <tr>
        <td><p>- Peça autorização ao Oficial de Quarto à Ponte para verificar os indicadores de ângulo de leme.</p></td>
        <td><%=estadoArray[2]%></td>
    </tr>

```

```

<tr>
  <td><p><strong>3.</strong> Junto ao próprio projector accione o comutador
  de alimentação, para estabelecer o projector. Pressionando o botão de
  alimentação a lâmpada interior acende?</p></td>
  <td><%=estadoArray[3]%></td>
</tr>
<tr>
  <td><p>No caso do projector não estar alimentado solicite a verificação do
  disjuntor de alimentação localizado no painel 3 do quadro eléctrico 2-39 e
  repita a execução do passo 3. </td>
  <td><%=estadoArray[4]%></td>
</tr>
<tr>
  <td><p>Alinhe de novo o projector na posição correcta e desligue o
  comutador de alimentação </p></td>
  <td><%=estadoArray[5]%></td>
</tr>
</table>

<p>Tempo padrão : 1:00</p>
<p>Data: <%=DataActual %> </p>
</body>
</html>

```

## Sailor:

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License

Name      : TwoColours
Description: A two-column, fixed-width design with dark color scheme.
Version   : 1.0
Released  : 20130811

-->
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.*" %>
<%@ page import = "java.util.*" %>

<html>
<head>

<meta name="keywords" content="" />
<meta name="description" content="" />
<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,70
0,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
<title> </title>

</head>

```

```

<body>

<div id="header">
    <div id="menu" class="container">
        <ul>

            <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
            <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
            <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
        </ul>
    </div>
</div>

<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2>    VHF DSC SAILOR    </h2>
        </div>
    </div>
</div>

<%

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;
String xt,xt2;
String DataAtual;
String estadoAtual="", duracaoSOC="", equipamento;
String estadoArray[];

estadoArray = new String[10];
int i = 0;
int j =0;

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
DataAtual = (new Date()).toString();

DataAtual = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();

xt = "select * from vhf_sailor; ";
xt2 = "select * from registo_tempo; ";
rs = st.executeQuery(xt);
rs = st.executeQuery(xt2);

while (rs.next())
{

```

```

String DataBD2;
DataBD2 = rs.getString("Data");
equipamento = rs.getString("Equipamento");

if(DataActual.equals(DataBD2) && equipamento.equals("VHF_DSC_Sailor")){
    duracaoSOC = rs.getString("Duracao");
}

}

while (rs.next())
{
    String DataBD;
    DataBD = rs.getString("Data");

    if(DataActual.equals(DataBD)){
        estadoActual = rs.getString("Estado");
        estadoArray[i]=estadoActual;
        i++;
    }

}

}

%>
<table width="725" border="1">
    <tr>
        <td width="592">Itens da SOC </td>
        <td width="117">Estado </td>
    </tr>
    <tr>
        <td><span class="style1"><strong>Nota: </strong>Para se realizar esta SOC
a impressora do DSC tem que estar ligada assim como o GPS <em>Trimble </em>(GPS
associado a este equipamento);</span> <br>
        <strong>(1)</strong> Estabeleça o equipamento, rodando o potenciômetro
para a direita.</td>
        <td><%=estadoArray[0]%></td>
    </tr>
    <tr>
        <td><strong>(a) </strong>Aparece no ecrã WATCHING CH 70 e hora ?</td>
        <td><%=estadoArray[1]%></td>
    </tr>
    <tr>
        <td><strong>(b) </strong>Confirme a hora com a informação dada pelo GPS
Trimble</td>
        <td><%=estadoArray[2]%></td>
    </tr>
    <tr>
        <td><br>
        <strong>(2) </strong>Pressione <strong>FUNC</strong>. Pressione " &gt; "
de forma a seleccionar a opção <strong>Position, </strong>prima<strong> NEXT;
a pst geográfica é idêntica à posição </strong>apresentada<strong> </strong>no
display<strong> do </strong>GPS <em>Trimble</em>?<strong> </strong></td>
        <td><%=estadoArray[3]%></td>
    </tr>

```

```

<tr>
  <td><br>
    <strong>(3)</strong> Pressione FUNC. Pressione em " &gt; " até a palavra
<strong>TEST</strong> ficar a piscar. Pressione <strong>NEXT</strong>.</td>
  <td><%=estadoArray[4]%></td>
</tr>
<tr>
  <td><br>
    <strong>(a) </strong>0 equipamento passou no teste? Pressione
<strong>NEXT</strong></td>
  <td><%=estadoArray[5]%></td>
</tr>
<tr>
  <td><strong>(4)</strong> Pressione em CALL. Seleccione FRAREAL. Pressione
em NEXT duas vezes. Pressione em SEND</td>
  <td><%=estadoArray[6]%></td>
</tr>
<tr>
  <td><strong>(a)</strong> O equipamento começa a apitar e dar informação de
mensagem recebida?</td>
  <td><%=estadoArray[7]%></td>
</tr>
<tr>
  <td><br>
    <strong>(b)</strong> A impressora imprime a informação da mensagem?</td>
  <td><%=estadoArray[8]%></td>
</tr>
<tr>
  <td><p><br>
    <strong>(c)</strong> Carregue em NEXT. Carregue em LAST duas vezes. </td>
  <td><%=estadoArray[9]%></td>
</tr>
</table>
<p>Tempo padrão: 5:00</p>
<p>Data: <%=DataActual %> </p>

</body>
</html>

```

## Sealand:

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License

Name      : TwoColours
Description: A two-column, fixed-width design with dark color scheme.
Version   : 1.0
Released  : 20130811

```

```

-->
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.*" %>
<%@ page import = "java.util.*" %>

<html>
<head>

<meta name="keywords" content="" />
<meta name="description" content="" />
<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,70
0,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
<title> </title>

</head>

<body>

<div id="header">
    <div id="menu" class="container">
        <ul>
            <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
            <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
            <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
        </ul>
    </div>
</div>

<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2> TX/RX VHF Sealand </h2>
        </div>
    </div>
</div>

<%

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;
String xt,xt2;
String DataActual;
String estadoActual="", duracaoSOC="", equipamento;
String estadoArray[];

estadoArray = new String[9];
int i = 0;
int j =0;

Class.forName("com.mysql.jdbc.Driver").newInstance();

```



```

conn = DriverManager.getConnection(url);

st = conn.createStatement();
DataActual = (new Date()).toString();

DataActual = ((new SimpleDateFormat("dd"/"/"MM"/"/"yy")).format(new
Date())).toString();

xt = "select * from vhf_sealand; ";
xt2 = "select * from registo_tempo; ";
rs = st.executeQuery(xt);
rs = st.executeQuery(xt2);

while (rs.next())
{
    String DataBD2;
    DataBD2 = rs.getString("Data");
    equipamento = rs.getString("Equipamento");

    if(DataActual.equals(DataBD2) && equipamento.equals("VHF_Sealand)){
        duracaoSOC = rs.getString("Duracao");
    }

}

while (rs.next())
{
    String DataBD;
    DataBD = rs.getString("Data");

    if(DataActual.equals(DataBD)){
        estadoActual = rs.getString("Estado");
        estadoArray[i]=estadoActual;
        i++;
    }

}

%>
<p>&nbsp;</p>
<table width="860" border="1">
    <tr>
        <td height="25">Itens da SOC </td>
        <td>Estado da SOC </td>
    </tr>
    <tr>
        <td width="689" height="25"><strong>1</strong> Estabeleça o equipamento,
colocando o comutador ON/OFF na posição ON.</td>
        <td width="155"><%=estadoArray[0]%></td>
    </tr>
    <tr>

```

<p><b>(a)</b> Aparece um número de canal no display alfanumérico?</p>	<%=estadoArray[1]%>
<p><b>(b)</b> O painel frontal é iluminado?</p>	<%=estadoArray[2]%>
<p><b>NOTA:</b> Caso não se verifiquem as duas alíneas anteriores, verifique o ajuste do DIMMER e/ou coloque o comutador BRIDGE WINGS em LOCAL.</p>	
<p><b>(2)</b> Ouve ruído de fundo?</p>	<%=estadoArray[3]%>
<p><b>NOTA:</b> Caso não ouça ruído de fundo, coloque o comutador E.L.U. ON/OFF/LOUDSPEAKER ON em LOUDSPEAKER ON, rode o potenciômetro de SQUELCH para a direita e ajuste o volume (potenciômetro VOLUME).</p>	
<p><b>(3)</b> Rode o potenciômetro SQUELCH para a esquerda até desaparecer o ruído de fundo.</p>	<%=estadoArray[4]%>
<p><b>(4)</b> Seleccione um canal, premindo a tecla CLEAR e o número do canal. Apareceu no display o número do canal seleccionado?</p>	<%=estadoArray[5]%>
<p><b>(5)</b> Seleccione a potência desejada (POT LOW/HIGH) e prima o PTT do Handset</p> <p>Acende-se a lâmpada indicadora de POT HIGH?</p>	<%=estadoArray[6]%>
<p><b>(6)</b> Verifique se o equipamento executa DUAL WATCH, procedendo da seguinte forma:</p> <p>Seleccione dois canais para DUAL WATCH, premindo as teclas A e CLEAR, seguido do número do canal.</p> <p>Seguidamente prima as teclas B e CLEAR, seguido do número do canal.</p> <p>Prima a tecla DUAL WATCH.</p> <p><b>(a)</b> No Display aparecem dois pontos a piscar alternadamente?</p>	<%=estadoArray[7]%>
<p><b>(b)</b> Logo que for recebido um sinal em qualquer destes canais, o display assinala o canal em uso?</p>	<%=estadoArray[8]%>

```

<p>Tempo padrão: 4:00</p>
<p>Data: <%=DataActual %> </p>

</p>
</body>
</html>

```

## Sereia:

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License

Name      : TwoColours
Description: A two-column, fixed-width design with dark color scheme.
Version   : 1.0
Released  : 20130811

-->
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.*" %>
<%@ page import = "java.util.*" %>

<html>
<head>

<meta name="keywords" content="" />
<meta name="description" content="" />
<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
<title> </title>

</head>

<body>

<div id="header">
    <div id="menu" class="container">
        <ul>

            <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
            <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
            <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
        </ul>
    </div>
</div>

```

```

<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2> Sereia </h2>
        </div>
    </div>
</div>

<%

java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;
String xt,xt2;
String DataAtual;
String estadoAtual="", duracaoSOC="", equipamento;
String estadoArray[];

estadoArray = new String[3];
int i = 0;
int j =0;

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
DataAtual = (new Date()).toString();

DataAtual = ((new SimpleDateFormat("dd"+"/"+"MM"+"/"+"yy")).format(new
Date())).toString();

xt = "select * from sereia; ";
xt2 = "select * from registo_tempo; ";
rs = st.executeQuery(xt);
rs = st.executeQuery(xt2);

while (rs.next())
{
    String DataBD2;
    DataBD2 = rs.getString("Data");
    equipamento = rs.getString("Equipamento");

    if(DataAtual.equals(DataBD2) && equipamento.equals("Sereia")){
        duracaoSOC = rs.getString("Duracao");
    }

}

while (rs.next())
{
    String DataBD;
    DataBD = rs.getString("Data");

```

```

        if(DataActual.equals(DataBD)){
            estadoActual = rs.getString("Estado");
            estadoArray[i]=estadoActual;
            i++;
        }

    }

%>
<p>&nbsp;</p>
<table width="750" border="1">
    <tr>
        <td width="658">Itens da SOC</td>
        <td width="76">Estado </td>
    </tr>
    <tr>
        <td>No painel de comando, verifique o seguinte:<br>
            <strong>(1)</strong> O potenciômetro de controle de intensidade luminosa
            actua? </td>

            <td><%=estadoArray[0]%></td>
        </tr>
        <tr>
            <td><br>
                <strong>(2)</strong> Pressione o botão superior esquerdo durante 1
                segundo. O sinal sonoro actua?</td>
            <td><%=estadoArray[1]%></td>
        </tr>
        <tr>
            <td><strong>(3)</strong> Pressione os botões existentes nas asas da Ponte e
            junto do VHF SEALAND. O sinal sonoro actua? </td>
            <td><%=estadoArray[2]%></td>
        </tr>
    </table>
<p>Tempo padrão: 1:30</p>
<p>Data: <%=DataActual %> </p>

</body>
</html>

```

### ConsultarHistorico:

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates

```

http://www.freecsstemplates.org  
Released for free under a Creative Commons Attribution 2.5 License

Name : TwoColours  
Description: A two-column, fixed-width design with dark color scheme.  
Version : 1.0  
Released : 20130811

```
-->
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.*" %>
<%@ page import = "java.util.*" %>

<html>
<head>

<meta name="keywords" content="" />
<meta name="description" content="" />
<link
href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700,900" rel="stylesheet" />
<link href="default.css" rel="stylesheet" type="text/css" media="all" />
<link href="fonts.css" rel="stylesheet" type="text/css" media="all" />
<title> </title>

</head>

<body>

<div id="header">
    <div id="menu" class="container">
        <ul>

            <li class="current_page_item"><a href="index.jsp"
accesskey="1" title="">Página Inicial</a></li>
            <li><a href="ConsultarHistorico.jsp" accesskey="1"
title="">Histórico</a></li>
            <li><a href="#" accesskey="5" title="">0
Desenvolvedor</a></li>
        </ul>
    </div>
</div>

<div id="page" class="container">
    <div id="content">
        <div class="title">
            <h2>    Histórico    </h2>
        </div>
    </div>
</div>

<%
String equi = "Adrica";
java.sql.Connection conn = null;
String url = "jdbc:mysql://localhost/test?user=root&password=tese";
java.sql.Statement st;
java.sql.ResultSet rs;

String equipamentoArray[],duracaoArray[], dataArray[];
```

```

equipamentoArray = new String[100];
duracaoArray = new String[100];
dataArray = new String[100];

int i = 0;
int j =0;

String xt; // variavel responsavel pela leitura da tabela (execute table)

Class.forName("com.mysql.jdbc.Driver").newInstance();
conn = DriverManager.getConnection(url);

st = conn.createStatement();
xt = "select * from registo_tempo;";
rs = st.executeQuery(xt);

while (rs.next())
{
    equipamentoArray[i] = rs.getString("Equipamento");
    duracaoArray[i] = rs.getString("Duracao");
    dataArray[i] = rs.getString("Data");
    i++;
}

rs.close();
conn.close();

%>

<%

String aux;

out.println(" <table border=1 cellpadding=4>");
out.println(" <tr><td><b>Durac o</b></td> <td><b>Equipamentos</b></td>");
out.println(" <td><b>Data da Realiza o</b></td></tr>");

for(i=0; i<equipamentoArray.length; i++)
{
    if(equipamentoArray[i] == null)
        break;
    else
    {
        out.println("<tr><td>" + duracaoArray[i] + "</td> " +
            "<td>" + equipamentoArray[i] + "</td> " +
            "<td>" + dataArray[i] + "</td> </tr>");
    }
}

```

```

}
}out.println("</table>");    %>

```

```

<form action="Pesquisa.jsp" method="post" >
    <div align="left">Data:
        <input type="text" name="data"/>
        exemplo: 18/06/2013 Equipamento:
        <input type="text" name="equipamento"/>
        (conforme a Tabela)
        <input type="hidden" name = "pd" value="Pesquisa"/>
        <input type="submit" value="Pesquisa"/>
        <span class="style1"> </span><br>
        <br>
    </div>
</form>

</body>
</html>

```

## Anexo F – Fichas de Testes das Aplicações

<b>Identificação</b>	T01
<b>Descrição</b>	Teste SOC OQP – equipamentos na responsabilidade do oficial de quarto à ponte.
<b>Pré-Condições</b>	Menu principal no ecrã do dispositivo; Registo de tempo padrão de realização de cada equipamento; Realizar a SOC de forma tradicional, em papel.
<b>Ação a Ser Tomada</b>	Realizar a SOC, validar os itens da aplicação ao executar as respetivas atividades.
<b>Resultado Esperado</b>	A aplicação deverá registar os tempos em que cada SOC foi realizada, ao terminar a SOC a aplicação envia a informação para o servidor com os tempos de cada SOC;  A aplicação deverá enviar para o servidor uma informação de estado de cada item da SOC (Falhou, ou Funciona Corretamente);  A Aplicação deverá enviar para servidor o nome do equipamento e a data da realização da SOC.
<b>Método para Validação</b>	O registo dos tempos está a funcionar corretamente; Aplicação estabelece ligação com servidor e envia os dados; O tempo decorrido é inferior ou igual ao tempo padrão.



<b>Identificação</b>	T02
<b>Descrição</b>	Teste SOC Sargento – equipamentos na responsabilidade do sargento de quarto à ponte.
<b>Pré-Condições</b>	Menu principal no ecrã do dispositivo; Registo de tempo padrão de realização de cada equipamento; Realizar a SOC de forma tradicional, em papel.
<b>Ação a Ser Tomada</b>	Realizar a SOC, validar os itens da aplicação ao executar as respetivas atividades.
<b>Resultado Esperado</b>	A aplicação deverá registar os tempos em que cada SOC foi realizada, ao terminar a SOC a aplicação envia a informação para o servidor com os tempos de cada SOC;  A aplicação deverá enviar para o servidor uma informação de estado de cada item da SOC (Falhou, ou Funciona Corretamente);  A Aplicação deverá enviar para servidor o nome do equipamento e a data da realização da SOC.
<b>Método para Validação</b>	O registo dos tempos está a funcionar corretamente; Aplicação estabelece ligação com servidor e envia os dados; O tempo decorrido é inferior ou igual ao tempo padrão.

<b>Identificação</b>	T03
<b>Descrição</b>	Teste SOC Marinheiro do Leme – equipamentos na responsabilidade do Marinheiro do Leme.
<b>Pré-Condições</b>	Menu principal no ecrã do dispositivo; Registo de tempo padrão de realização de cada equipamento; Realizar a SOC de forma tradicional, em papel.
<b>Ação a Ser Tomada</b>	Realizar a SOC, validar os itens da aplicação ao executar as respetivas atividades.
<b>Resultado Esperado</b>	A aplicação deverá registar os tempos em que cada SOC foi realizada, ao terminar a SOC a aplicação envia a informação para o servidor com os tempos de cada SOC;  A aplicação deverá enviar para o servidor uma informação de estado de cada item da SOC (Falhou, ou Funciona Corretamente);  A Aplicação deverá enviar para servidor o nome do equipamento e a data da realização da SOC.
<b>Método para Validação</b>	O registo dos tempos está a funcionar corretamente; Aplicação estabelece ligação com servidor e envia os dados; O tempo decorrido é inferior ou igual ao tempo padrão.

<b>Identificação</b>	T04
<b>Descrição</b>	Teste SOC Equipamentos – Consulta do estado dos equipamentos.
<b>Pré-Condições</b>	Página principal no ecrã do computador; Registo de tempo padrão de realização de cada equipamento; Realizar a SOC de forma tradicional, em papel.
<b>Ação a Ser Tomada</b>	Fazer a consulta de todas as páginas, cujos equipamentos foram testados na aplicação móvel
<b>Resultado Esperado</b>	A aplicação deverá ser capaz de consultar os dados enviados da aplicação móvel ao servidor.
<b>Método para Validação</b>	Os dados enviados condizem com o que foi efetuado na aplicação móvel; Aplicação estabelece ligação com servidor e consulta os dados; É possível fazer a consulta do histórico.